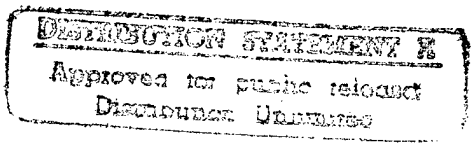
Machine Translation of Battlefield Messages by Lexico-Structural Transfer

Scientific and Technical Report
for SBIR Phase I Project



Final Version, July 30, 1997

SCIENTIFIC AND TECHNICAL REPORT
for
SBIR Phase I Project
"Machine Translation of Battlefield Messages by Lexico-Structural Transfer"

CONTRACT NO. DAAL01-97-C-0016
CDRL No. A001

Prepared for:
Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

Prepared by:
CoGenTex, Inc.
840 Hanshaw Road, Suite 5
Ithaca, NY 14850
Contact: Owen Rambow
owen@cogentex.com

With the University of Pennsylvania, Subcontractor

Contents

Chapter 1

Introduction

This report presents the results of a Phase I SBIR funded by the Army Research Laboratory entitled "Machine Translation of Battlefield Messages by Lexico-Structural Transfer" (contract DAAL01-97-C-0016). The goal of this Phase I effort has been to explore ways of automatically translating battlefield messages. More specifically, the aims of the project have been as follows:

- Identify requirements and opportunities specific to the application domain.
- Show the feasibility of using software developed previously at CoGenTex and at the University of Pennsylvania in order to quickly assemble an MT system which addresses the specific requirements of battlefield messages.
- Identify the principal areas which require further work during a subsequent effort in order to transform the Phase I feasibility demonstration into a fully functional prototype system.

In this report, we describe how we have achieved these aims.

This report combines the four main deliverables required under our SBIR "Machine Translation of Battlefield Messages by Lexico-Structural Transfer" (contract DAAL01-97-C-0016), in a single document for the reader's convenience.

Specifically, this report contains the following chapters.

- **Chapter 2** (page 5) provides an overview of the system and introduces the key level of lingusitic representation.
- **Chapter 3** (page 17) is a short report on the sublanguage corpora and characteristics for the two chosen sublanguages.

- **Chapter 4** (page 19) is the report discussing and justifying the choice of parser, and discussing the process of specializing the parser for the sublanguage of the Battlefield Message domain.
- **Chapter 5** (page 36) is the report discussing and justifying the choice of transfer formalism, and discussing which types of translation divergences can be handled how.
- **Chapter 6** (page 55) is a short report discussing the process of specializing REALPRO, CoGenTex's sentence realization system, for the sublanguage of the Battlefield Message domain, and discussing the issue of transfer component/generator gaps.
- **Chapter 7** (page 58) is an additional report (not required by the contract) which summarizes how this system can be ported to new language pairs and to new domains, based on our experience porting the system to English/Arabic translation.

Throughout this report, we will evaluate the effort in sections entitled "Discussion".

Chapter 2

Overview of the System

2.1 Background and Requirements

Military action has always involved forces from more than one country, but with the end of the Cold War, new coalitions of forces have emerged in military engagements throughout the world. These include countries that formerly belonged to the Warsaw Pact, or countries around the world that join with the United States in regional military action. The result is that military personnel from more countries need to communicate with each other than before, and, furthermore, that the specific communication needs may become apparent only at short notice. There is therefore a great practical need for the automatic translation of battlefield messages from one natural language to another, i.e., for machine translation (MT).

We have identified the following three principal requirements for MT of battlefield messages:

- The system must robustly translate relatively short messages in several subdomains of the battlefield message domain.
- The system must run on standalone PCs in operational contexts.
- The system must be easily portable to new language pairs and to new subdomains.

MT has been, almost from the beginning of electronic computing, an arduously pursued goal of much research in academia and the commercial world. However, the task quickly turned out to be far more complex than originally anticipated. While there is clearly a very large demand for MT in the world in many contexts, current technology still cannot meet the broad requirements found in many domains. However, battlefield messages have several distinguishing characteristics:

- The language used, while not always standard language (telegraphic style), can be represented as a sublanguage.

- Language-internal ambiguities are limited within the sublanguage due to speaker awareness of sublanguage requirements.
- The domains of discourse are limited in number and relatively well defined.

While free-text, domain-independent, high-quality MT remains beyond the state of the art today, this project makes use of several recent developments which exploit the specific characteristics of battlefield messages and which make it possible to attain the requirements set forth above. Specifically, we exploit the following scientific advances:

- The statistical study of sublanguages has progressed to the point at which efficient techniques are available for quickly analyzing linguistic communication in restricted domains.
- Broad-coverage, well-tested natural language software is now available for both analysis and generation.
- Statistical methods have been developed which make it easy to configure some MT component software to new domains and language pairs.
- Standardized, powerful, and mobile computing capability is now widely available at low cost (PC).

In this project, CoGenTex, Inc., and the University of Pennsylvania leverage cutting-edge academic research into a usable application tailored to the needs of the Army Command and Control system domain.

2.2 Approach

In this project, we use an approach to MT based on the notion of *lexico-structural transfer*. This approach has two main characteristics.

- The approach is lexicalist.
- The approach is hybrid, encompassing both linguistic and stochastic methods.

We will explain these points in more detail.

In lexico-structural transfer, we do not use a separate level of representation which would be intended as a truly language-independent representation of the meaning of the sentence and would traditionally be termed an "interlingua". Instead, our lexicalized grammar approach provides us with a unified syntactic and semantic representation for each lexical item. The dependency relations we derive during the parsing process (the DSyntS – see Section 2.4

below) represent directly the predicate-arugment structure; the DSyntS can in addition be richly annotated with semantic features from the lexicon. By including appropriate crosslinguistic semantic features, and coindexing them in the transfer lexicon, we can capture the same generalizations that are traditionally associated with an interlingua approach, without requiring a separate level of representation (Palmer and Rosenzweig, 1996).

Instead, we have found that a syntactic dependency structure annotated with semantic features provides a level of representation that allows for an easy encoding of a transfer lexicon and yet is also rich enough to allow generalizations based on cross-linguistic features (Palmer and Rosenzweig, 1996). This provides an elegant and efficient method of grouping together in the transfer lexicon entire classes of lexical items that are structurally divergent in the source language and target language (in the sense of (Dorr, 1994)), yet in a predictable fashion. The lexicalized approach also allows for a fine-grained treatment of frozen and semi-frozen expressions such as idioms. We can thus sidestep some of the difficult issues involved in defining, deriving and motivating a "true interlingua" while still retaining the benefits of a representation that can readily be mapped onto many languages.

An additional advantage of a lexically-based transfer approach is that it is still fairly close to the surface structure. This allows us to exploit statistical techniques for analyzing corpora and for extracting information from them (including translation lexicons). (This is notably difficult in interlingua-based approaches, where the interlingua is of course never manifest, but rather a construction of the researchers.) We have extracted large parts of the translation lexicons for our subdomains automatically, and we have trained two different parsers on the syntactic structures in the corpora to improve their performance.

The statistical techniques, of course, have limited applicability. Machine translation is a notoriously complex task consisting of many steps (tagging, parsing, transfer, generation). Error margins in a single step are compounded in combination, so that acceptable error margins for a standalone parser, say, may not be acceptable for a parser operating as part of an MT system. Therefore, we have designed our system in such a way that statistically derived information can be easily complemented by linguistic information hand-encoded by linguists. This is possible because of our lexicalist approach: the linguistic knowledge that has been added by hand is related to specific lexical items or classes of lexical items, so that the hand-coded knowledge can easily interact with the statistically derived knowledge.

We are also adhering to the current acceptance of a modular architecture which allows modules to be readily exchanged. We define interface requirements for the individual modules which could be satisfied by implementations based on competing theoretical approaches. This allows us to compare and contrast alternative implementations and alternative theories, and to assemble solutions for particular MT tasks from existing linguistic resources.

In our feasibility study system, we have used two parsers previously developed at Penn and a generator previously developed at CoGenTex. These components were developed for different purposes (i.e., not necessarily for MT), and they are based on different theories of language, but they all share a lexicalist approach. We could substitute other lexicalist components, such as a synchronous TAG-based system, if they promised performance improvements. We

have chosen the particular components we use because of their relative maturity as software components.

2.3 Overview of the System

Skies were clear across the three maritime provinces early this morning.
$\longrightarrow$ Le temps était clair dans les trois provinces maritime ce matin tôt.
Behind this area a moderate flow will cause an inflow of milder air in southwestern Quebec producing mild temperatures on Sunday.
$\longrightarrow$ Une circulation modérée provoquera un afflux du air doux dans le sud-ouest du Québec à l'arrière de cette zone produisant des températures dimanche douxes.
Loyalty of local civilian officials is questionable.
$\longrightarrow$ La loyauté des dirigeants locaux civils est douteuse.
The 175tr/9gtd is moving west on e4a48 Autobahn toward Berlin.
$\longrightarrow$ Le 175tr/9gtd se déplace vers l'ouest sur e4a48 autobahn vers Berlin.

Figure 2.1: Some sample translations performed by TransLex

This is an overview of our feasibility prototype machine translation system, which we will call TransLex. TransLex is an English-to-French translation system. Some sample outputs can be seen in Figure 2.1. The main level of representation in TransLex is a syntactic dependency representation which we will call DSyntS, for *Deep Syntactic Structure* (roughly as defined in (Mel'čuk, 1988)). This level of representation contains all the meaning-bearing words of a sentence (nouns, verbs, adjectives, adverbs, and some prepositions, but no auxiliary verbs, strongly governed prepositions, and so on), and relates them syntactically using a small set of possible relations (essentially, arguments and adjuncts). We discuss the DSyntS in more detail below (Section 2.4, page 10). The DSyntS is closely related to the *derivation structure* of Tree Adjoining Grammar; see (Rambow and Joshi, 1996) for details.

TransLex consists of the following components:

- Two parsers (Bilder and the SuperTagger from the University of Pennsylvania), each with a converter which converts the output from the parser to the DSyntS.
- The core transfer component.
- The generator (RealPro).

The architecture is shown in Figure 2.2. The role of the two parsers is currently to show that parsing is possible to a sufficient level of precision; in an operational prototype we will either choose one or the other, or find a way of combining their output for the sake of optimization.

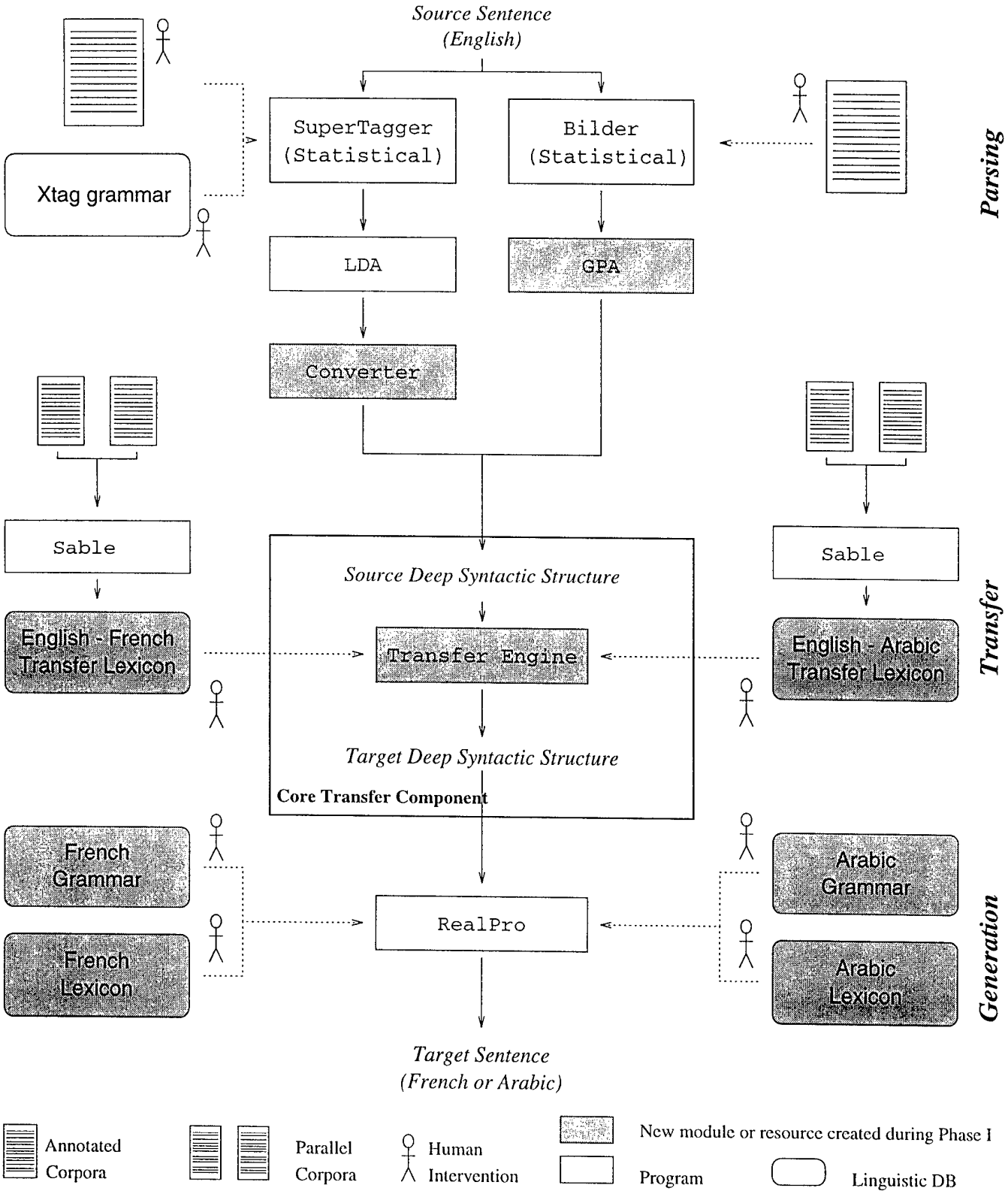


Figure 2.2: TransLex system architecture

2.4 DSyntS: The Linguistic Representation in TransLex

2.4.1 The Linguistic Representation in TransLex

The linguistic representation in TransLex is a syntactic dependency representation. It is called the Deep-Syntactic Structure or "DSyntS" for short, and is proposed in this form by I. Mel'čuk in his Meaning-Text Theory (Mel'čuk, 1988). This representation has the following salient features:

- The DSyntS is a *tree* with labeled nodes and labeled arcs.
- The DSyntS is *lexicalized*, meaning that the nodes are labeled with lexemes (uninflected words) from the target language.
- The DSyntS is a *dependency* representation and not a *phrase-structure* representation: there are no nonterminal nodes (such as VPs), and all nodes are labeled with lexemes.
- The DSyntS is a *syntactic* representation, meaning that the arcs of the tree are labeled with syntactic relations such as "subject", rather than conceptual (or "semantic") relations such as "agent".
- The DSyntS is a *deep* syntactic representation, meaning that only meaning-bearing lexemes are represented, and not function words.

This means that the linguistic representation specifies all meaning-bearing lexemes.

2.4.2 The Notion of Syntactic Dependency

Syntactic dependency is a fundamental notion about the syntax of natural (human) languages that itself cannot be derived from more primitive notions. Intuitively, in a sentence a lexical item l_1 is dependent on another lexical item l_2 if l_1's presence in the sentence is only licensed by the presence of l_2: put differently, l_1 may be present in the sentence because l_2 is also present. There are, broadly speaking, two types of (deep) dependency:

- *Complementation* is the relation between a lexeme which is a predicate and one of its arguments, i.e., a lexeme for which it selects. For example, in *John sneezes*, *to sneeze* selects for *John* as it subject (to express the conceptual role of, say, agent). (In English, subjects must usually be expressed overtly, and **sneezes* is not in itself a correct utterance.) Without an occurrence of *sneezes* (or any other verb) that selects for it, *John* is not licensed in a sentence. For example, the strings *John Mary sneezes* and *Mary sneezes John* are not valid sentences in English because *John* has nothing to depend on, i.e., is not licensed. Complementation relations correspond to the traditional notions of subject, object, indirect object, and so on.

- *Modification* is the relation between a head lexeme and a lexeme which is not an argument, but an adjunct. For example, in *I like small donuts*, *small* modifies *donuts*, and omitting *donuts* leads to an incomplete sentence, since then *small* has nothing to depend on and is no longer licensed: * *I like small.*

Thus, we only represent syntactic relations that are semantically meaningful, and not those that are the results of surface syntactic configurations. For example, for *who do you like?* the DSyntS shows the (deep-syntactic) relation between *who* and *like* (namely, *who* is the object of *like*), but not the (surface-syntactic) relation between *who* and *do*. In fact, the *do* does not even appear in the DSyntS, since it is just a function word and does not carry meaning on its own. Furthermore, we assume that each lexical item in a sentence depends on exactly one other lexical item, and that there are no cycles of dependency. For example, this means that if two verbs share a single overt argument through control (such as in *John promised Mary to leave*, where *John* is the subject both of *promise* and of *leave*), only one of the verbs (namely, *promise*) can have the argument as an overt dependent in the DSyntS. Of course, we then must assume that there is a single lexical item that does not depend on any other lexical item, i.e., the root of the dependency tree. In full sentences, this will always be the main verb of the sentence (though of course DSyntSs need not contain a verb).

Note that the notion of syntactic dependency is semantically meaningful, it is not the same as the notion of semantic dependency (as it is often represented in theories of formal semantics). While a sentence such as *John sneezes* is typically represented semantically as something like sneeze(John), i.e., with the same dependency as in the syntactic representation, this is not the case for modification: *small donuts* is typically represented semantically as small(donut), while syntactically *small* depends on *donut*. Thus, while syntactic dependency represents predicate-argument structure, it does not directly translate into a predicate-calculus-type representation of modification.

2.4.3 Quick Guide to the DSyntS

2.4.3.1 Nodes

Nodes are labeled with lexemes or lexical functions. A lexeme can either be in the lexicon, or not. It may also be fictitious, i.e., it need not correspond to a realizable lexeme in the language. (In this case, further processing is needed prior to realization.)

Each lexeme is accompanied by a list of features. If a lexeme is not in the lexicon, feature class must be specified. If a lexeme has complex irregular morphology or complex irregular syntactic behavior, it is recommended to add it to the lexicon.

Note that lexemes can be phrasemes (idioms). For example, in *John kicks the bucket*, *kicks the bucket* is represented by a single node (annotated, say, KICK_THE_BUCKET). The entry for this lexeme in the lexicon then contains the appropriate expansion at the next level of representation.

Lexical functions are functions from lexemes to sets of lexemes which express certain meaningful regularities within the lexicon of a given language. The standard example is magn, which expresses the notion of magnification for a noun. For example, magn(rain) is *heavy*, but magn(pluie) is *forte* 'strong'. Thus, lexical functions are used when the choice of lexeme is not in itself meaningful, but rather represents a conventionalized meaning ("magnification") and whose realization depends on another node in the tree (in this case the mother node, *rain* or *pluie*). It is clear that for translation, the lexeme should not be translated literally. See Section 5.4.2.1 for details.

2.4.3.2 Relations

The following DSynt relations can be used to connect nodes.

- Complementation: I , II , III, IV.

  These correspond to subject, direct object, indirect object, and additional complement (some English verbs take four complements, such as *bet* in *I bet John four dollars that he could not jump over the fence*). Every node may have at most one complement of each type.

- Modification: ATTR, DESC-ATTR

  A node may have zero or more modifiers.

- Miscellaneous: COORD, APPEND

  A node may have only one dependent connected to it by one of these relations.

2.4.3.3 Some Examples

In the following, we will be representing syntactic trees as lying on their sides. For example, the upper tree in Figure 2.3 will be represented as the lower tree.

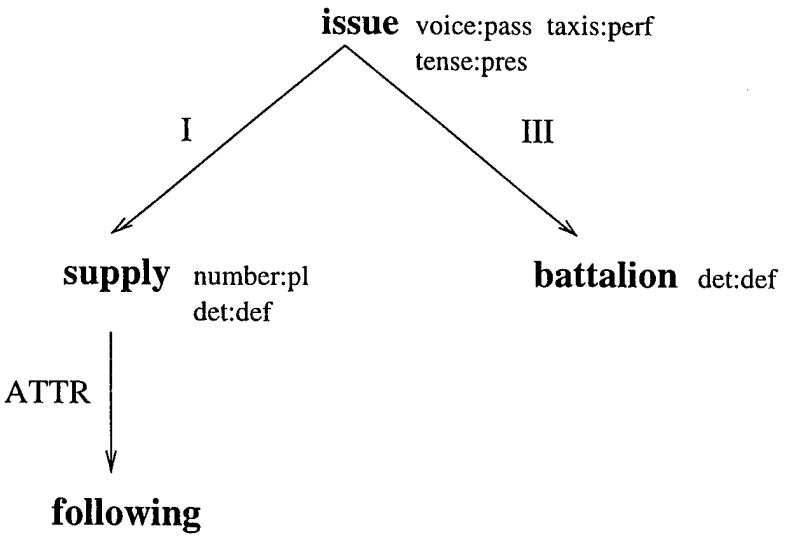
Here is an example of a clause. A clause is headed by a verb. We will write the head first, with its dependents following between parentheses. For clarity, each dependent will be on a new line. Before the name of each dependent node we will write the name of the relation.

The verb's arguments (subject, direct object, indirect object) are specified using the arc labels I, II, III. Of course, an intransitive verb only has a subject, while a transitive verb only has a subject and an object.

Input:

```
DSYNTS:

like [ class:verb tense:pres ]
```




```
DSYNTS:

issue [ class:verb tense:pres taxis:perf voice:pass ]
( I   supply   [ class:proper_noun number:pl article:no-art ]
         ( ATTR following [ class:adjective ] )
  III battalion   [ class:proper_noun article:def ] )
)
```

Figure 2.3: DSyntS tree for *The following supplies have been issued to the battalion.* in standard representation (above) and in "lateral" representation (below)

```
( I  Mary    [ class:proper_noun ]
  II John    [ class:proper_noun ] )
)
```

Output:

```
Mary likes John.
```

Any number of verbal features can be added to the verb, as can adverbs.

Input:

```
DSYNTS:
harass [ class:verb tense:past taxis:perf polarity:neg aspect:cont mood:ind ]
( I  Mary    [ class:proper_noun ]
  II John    [ class:proper_noun ]
  ATTR really [ class:adverb ]
)
```

Output:

```
Mary had not really been harassing John.
```

Verbs need not be finite. A verb can also take a clause as a complement. If there is no lexical entry for the matrix verb, the mood and tense (if applicable) of the embedded verb need to be specified. Example input:

```
BE1 [ mood:cond ]
( I  like [ class:verb mood:inf-to ]
     (  I   John [ class:proper_noun ]
        II  Mary [ class:proper_noun ]
     )
  II problem [ class:common_noun article:indef ]
)
```

Output:

```
For John to like Mary would be a problem.
```

Note that in the above example, BE1 is a lexeme which is defined in the lexicon (since it has irregular morphology).

Relative clauses are formed by adding the relative clause to the modified noun using the relations ATTR (for restrictive relative clauses) or DESC-ATTR (for descriptive relative clauses). Example input:

```
detest [ class:verb tense:past ]
( I    John [ class:proper_noun ]
  II   painting [ class:common_noun ref:pt-12 number:pl ]
       (  DESC-ATTR produce [ class:verb taxis:perf tense:past ]
            (  I    great_aunt  [ class:common_noun article:indef ]
               II   painting [ class:common_noun ref:pt-12 ]
                    ( ATTR ALL [ ]
                    )
            )
          ATTR <POSSESSIVE_PRONOUN> [ number:sg person:3rd gender:masc ]
    )
)
```

Output:

```
John detested his paintings, all of which a great aunt had produced.
```

Note that descriptive relative clauses are included between commas. Special processing eliminates the second comma in the presence of a period (or certain other punctuation marks). Note also that in the above example, the possessive pronoun *his* is generated from an abstract lexeme <POSSESSIVE_PRONOUN> with the appropriate feature markings. This eliminates the need to choose correct pronominal forms in the input specification.

Clauses and nouns can be coordinated using the COORD relation. Example input:

```
laugh [ class:verb tense:past ]
( I      John [ class:proper_noun ]
  COORD  BUT [ ]
         ( II  smack [ class:verb tense:past ]
               (  I   Mary    [ class:proper_noun ]
                  II  butler  [ class:common_noun article:def ]
                      ( COORD AND2  [ ]
                              (  II  maid [ class:common_noun article:def ]
                               )
                      )
```

```
                    )
                )
    )
```

Output:

```
John laughed but Mary smacked the butler and the maid.
```

Chapter 3

Sublanguage

This chapter is a short report on the characteristics of the sublanguag corpora used in this effort.

In order to be able to train the components of our MT system, we used two corpora:

- The CECOM battlefield message corpus.
- The Montreal weather forecast corpus.

We will discuss them in more detail.

3.1 Characteristics of The Battlefield Message Corpus

The CECOM battlefield message corpus was collected during a training exercise held at Ft. Leavenworth, Kansas (controlled by the 75th Brigade in Houston). It was subsequently translated by civilian translators in Canada into French. The size of the corpus is 7302 words, 48.5 Kilobytes of data. We divided it into a training corpus of 5551 words and a test corpus of 1751 words.

This corpus has the following characteristics:

- There are several subdomains within the corpus, notably weather forecasts, troop movements and locations, and meta-communication (communication about the communication, for example *checking transmission*).
- In most subdomains, there are requests for information and the corresponding answer (typically, a type of status report).
- There are many short sentences and fragmentary sentences, though standard English syntax is also found (especially in the meta-communication subdomain).

- There is a high percentage of jargon and abbreviations.

An exhaustive analysis of this corpus can be found in (Bourbeau, 1991).

3.2 Characteristics of The Weather Corpus

The Montreal weather forecast corpus collected at the Université de Montréal by Richard Kittredge. This is a bilingual corpus of texts issued by Environment Canada, with English and French texts provided (such that one is translated from the other by a human). This corpus has also been translated into Arabic (see Section 7). The size of the corpus is 3970 words, 23.7 Kilobytes of data. We divided it into a training corpus of 2914 words and a test corpus of 1056 words (50 sentences). We have used this corpus since it is an extension of one of the subdomains of the battlefield message corpus.

This corpus has the following characteristics:

- The domain is very consistent, and the number of lexical items (types) is quite small.
- The sentences are in standard English syntax and are often quite long (an average of 20 words), with participial adjuncts and conjunctions.
- There is a fairly high percentage of domain-specific terminology, but there are no or few abbreviations.

As can be seen, the two corpora are quite different, and we have therefore retained both of them for our study since they provide rather different test cases for our approach.

Chapter 4

The Parsers

4.1 Introduction: Two Parsers

In this project, we have investigated the use of two parsers, both developed previously at Penn, namely Bilder (Collins, 1996) and the SuperTagger with Lightweight Dependency Analysis (Joshi and Srinivas, 1994; Srinivas, 1997). These parsers are rather different:

- Bilder is trained on a corpus annotated with phrase-structure parse trees, and uses the probability of specific word-word dependencies to determine the most likely parse.
- The SuperTagger is trained on a corpus annotated with an expanded part-of-speech set ("supertags"). It uses only these supertags to heuristically determine the most likely parse.

The SuperTagger therefore may be easier to train for new domains or new languages, but Bilder may be better equipped to deal with certain types of attachment ambiguities. As part of this effort, we have trained both parsers on the corpora of interest to us.

Neither parser produces an output in the format which we need as the input for our transfer module, the DSyntS (see Section 2.4, page 10 for details). Therefore, both parsers use "converters". Bilder, which outputs a phrase-structure parse tree annotated with head information, uses the Generic Parse Analyzer (GPA) developed at Penn, which has been specialized for outputting a DSyntS during this project (Section 4.2.2, page 23). The SuperTagger/LDA outputs a dependency tree which is based on the derivation structure of Tree Adjoining Grammar; while this representation is very close to the DSyntS, it is not identical (see (Rambow and Joshi, 1996)). As part of this project, we have added a small converter to bridge the gap (Section 4.3.5, page 32).

For this feasibility study, we have chosen to use both parsers, and to evaluate their performance. We will, in a subsequent project, develop a model that uses one or the other or both parsers in order to optimize the results.

4.2 Bilder and the GPA

4.2.1 Bilder

4.2.1.1 Statistical Parsing

Recently, statistical parsers (e.g. (Magerman, 1995b; Collins, 1996)) have been shown to be highly effective at parsing unrestricted text in domains such as the Wall Street Journal. Such parsers are trained on a set of (sentence, tree) pairs, and will then output the most likely parse for a new, novel, sentence. One advantage of these methods is that they offer a principled solution to the problem of *ambiguity* – competing analyses for a sentence can be ranked in order of probability. This is important considering that, in practice, there can be a combinatorial explosion in the number of analyses for a sentence. For example, statistical methods have been used to resolve prepositional-phrase attachment ambiguity (e.g. (Collins and Brooks, 1995)) with around 85% accuracy, a surprisingly good result on a "hard" problem.

We have used Bilder (described fully in (Collins, 1996)) as one component in this project. Bilder bases its statistical model on co-occurrences between head-words in parse trees – essentially a generalization to full parsing of the lexicalized approach to pp-attachment (Collins and Brooks, 1995). Bilder recovers constituents in Wall Street Journal with over 85% accuracy, when trained on the Penn WSJ treebank (Marcus et al., 1993). The probabilistic information about partial analyses allows aggressive pruning of the parser search space, giving parsing speeds of around 200 sentences per minute.

4.2.1.2 What Bilder Does

Lexical information has been shown to be crucial for many parsing decisions, such as prepositional-phrase attachment (for example (Hindle and Rooth, 1993)). However, early approaches to probabilistic parsing (Pereira and Schabes, 1992; Magerman and Marcus, 1991; Briscoe and Carroll, 1993) conditioned probabilities on non-terminal labels and part of speech tags alone. The SPATTER parser (Magerman, 1995b; Jelinek et al., 1994) does use lexical information, and recovers labeled constituents in Wall Street Journal text with above 84% accuracy – as far as we know the best published results on this task. Bilder is much simpler than SPATTER, yet performs at least as well when trained and tested on the same Wall Street Journal data. Bilder uses lexical information directly by modeling head-dependent relations between pairs of words. In this way it is similar to Link grammars (Lafferty et al., 1992), and dependency grammars in general.

The aim of a parser is to take a tagged sentence as input (for example Figure 4.1(a) or Figure 4.2(a)) and produce a phrase-structure tree as output (Figure 4.1(b) or Figure 4.2(b)). A statistical approach to this problem consists of two components. First, the *statistical model* assigns a probability to every candidate parse tree for a sentence. Formally, given a sentence

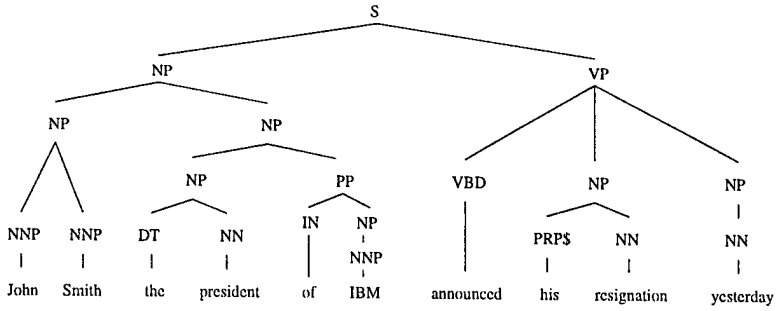
S and a tree T, the model estimates the conditional probability $P(T|S)$. The most likely parse under the model is then:

$$T_{best} = argmax_T P(T|S) \tag{4.1}$$

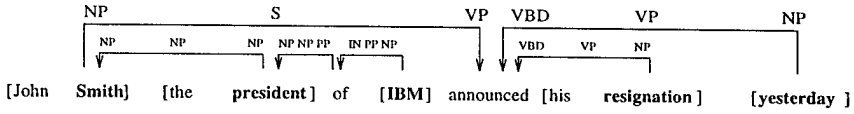
(a)

John/NNP Smith/NNP, the/DT president/NN of/IN IBM/NNP, announced/VBD his/PRP$ resignation/NN yesterday/NN .

(b)



(c)



(d)

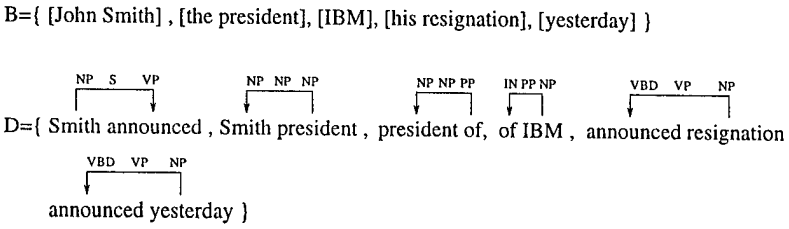


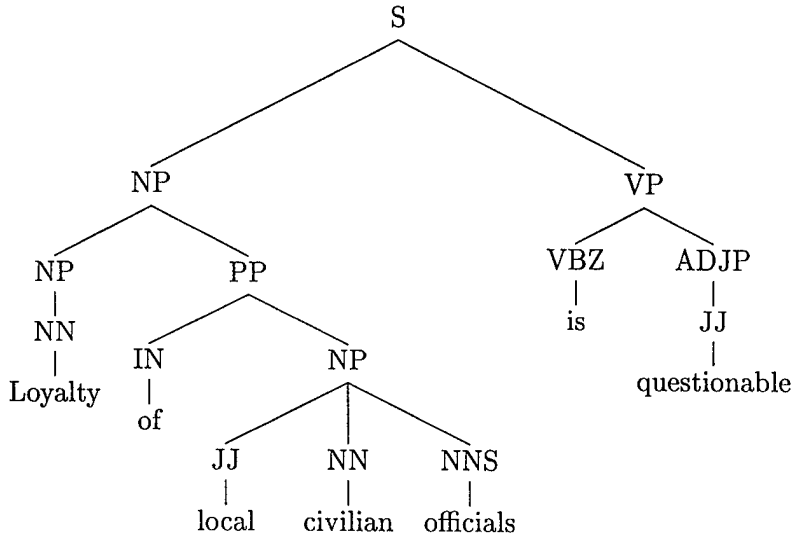
Figure 4.1: Example 1. **(a)** The tagged sentence; **(b)** A candidate parse-tree (the correct one); **(c)** A dependency representation of (b) (square brackets enclose baseNPs, heads of baseNPs are marked in bold, arrows show modifier $\rightarrow$ head dependencies). **(d)** B, the set of baseNPs, and D, the set of dependencies, are extracted from (c).

Second, the *parser* is a method for finding T_{best}. For example a simple (but hopelessly inefficient) parser would follow a generate and test procedure – it would enumerate every possible tree for a sentence, rank them using $P(T|S)$, and finally identify T_{best} as the top-ranking tree. Instead, Bilder uses a more sophisticated parser which overcomes these problems.

(a)

Loyalty/NN of/IN local/JJ civilian/NN officials/NNS is/VBZ questionable/JJ

(b)



(c)

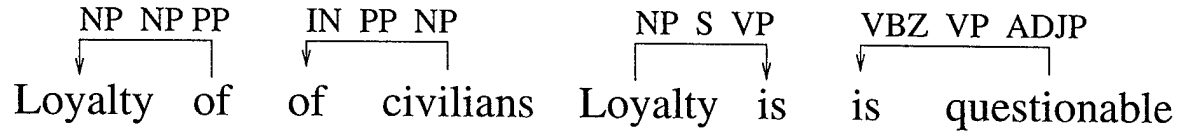


Figure 4.2: Example 2. **(a)** The tagged sentence; **(b)** A candidate parse-tree (the correct one); **(c)** A dependency representation of (b).

4.2.1.3 Evaluating Bilder

Prior to the use in this project, the parser was trained on sections 02 - 21 of the Wall Street Journal portion of the Penn Treebank (Marcus et al., 1993) (approximately 40,000 sentences), and tested on section 23 (2,416 sentences). For comparison, SPATTER (Magerman, 1995b; Jelinek et al., 1994) was also tested on section 23. We use the PARSEVAL measures (E. Black et al., 1991) to compare performance:

$$\textbf{Labeled Precision} = \frac{number\ of\ correct\ constituents\ in\ proposed\ parse}{number\ of\ constituents\ in\ proposed\ parse}$$

$$\textbf{Labeled Recall} = \frac{number\ of\ correct\ constituents\ in\ proposed\ parse}{number\ of\ constituents\ in\ treebank\ parse}$$

Crossing Brackets = number of constituents which violate constituent boundaries with a constituent in the treebank parse.

For a constituent to be 'correct' it must span the same set of words (ignoring punctuation, i.e. all tokens tagged as commas, colons or quotes) and have the same label[1] as a constituent in the treebank parse. Four configurations of the parser were tested: **(1)** The basic model; **(2)** The basic model with a punctuation rule; **(3)** Model (2) with tags ignored when lexical information is present; and **(4)** Model (3) also using the full probability distributions for POS tags. We should emphasise that test data outside of section 23 was used for all development of the model, avoiding the danger of implicit training on section 23.

For a description of the result of training Bilder on the battlefield message corpus, see Section 4.2.3, page 25.

4.2.2 GPA

The Generic Parse Analyzer (GPA) is a module for robust, domain-independent extraction of semantically relevant lexical relationships from syntactic annotations of texts. The module accepts parse trees as its input and returns as its output an approximation of a logical semantic form with detailed lexical predicates. The module draws on a lexical semantic knowledge base including information about verb subcategorization, the ontology of noun-phrase referents, and compound lexical items spanning multiple words of text. It combines this knowledge source with an encoding of existing linguistic knowledge about comparatively well-understood, basic features of the syntax-semantics interface. The GPA module is efficiently implemented in C.

The module is designed to be flexible enough to accept a wide range of parse-tree annotation styles as its input, so long as these conform to a general schema reflecting uncontroversial, mainstream theories of syntax. The module does not require, but may benefit from, comparatively rich annotation schemes which indicate the presence of empty constituents and/or movement, the thematic type of verb complements, and other such information.

It has initially been tailored to process annotations in the style of the original Penn Treebank project, since there is currently no other set of human-annotated parse trees as large and varied as the original Treebank. However, it is also capable of processing the output of statistical parsers trained on the Penn Treebank, even though the annotation supplied by such parsers is usually more impoverished. The availability of large sets of such automatically generated parse trees opens up the possibility for statistical refinement of the parse-tree analysis module through unsupervised learning techniques. By coupling a statistical parser such as the one described in (Collins, 1996) with the GPA module in series, useful predicate-argument relations can be extracted automatically from raw input text. It is also possible

[1]SPATTER collapses ADVP and PRT to the same label, for comparison we also removed this distinction when calculating scores.

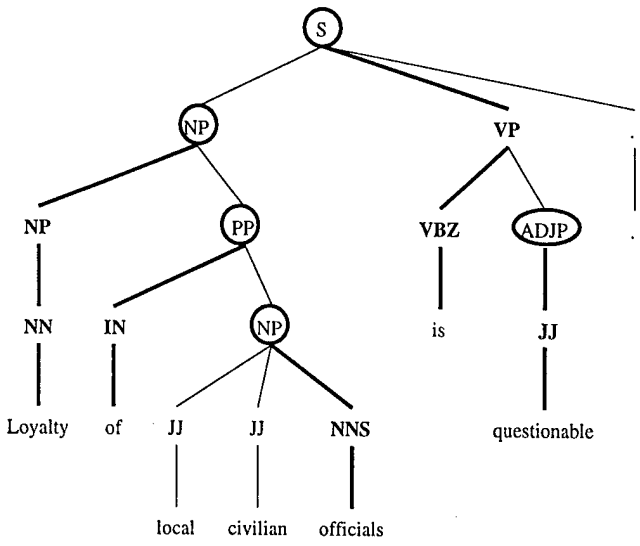


Figure 4.3: Phrase structure analysis provided by Bilder for sentence *Loyalty of local civilian officials is questionable.*; bold lines indicate projection from lexical items to their maximal projections

that both modules might improve their performance if they were able to share some of their now separate knowledge in a concurrent processing framework.

The output of the module is patterned after, but supersedes, such formalisms as the Intermediate Syntactic Representation of the PUNDIT language understanding system (Hirschman et al., 1989), the semantic side of the synchronous TAG transducer of Shieber and Schabes (Shieber and Schabes, 1990), or quasi-logical forms. It is intended to serve as a linguistically motivated predicate-argument annotation of text for the purposes of information extraction and machine translation. Its output is compatible in a straightforward way with standard conceptions of discourse models, so that the GPA analyses can be used to incrementally update a knowledge base that tracks the contextual information in a text as it is being processed. Such analyses, coupled with a suitable model of discourse context, are important for tasks such as template filling from text databases. The GPA output is also compatible with DSyntS, the transfer representations used in TransLex. The DSyntS can be seen as a more syntactic version of the logical semantic form that GPA normally outputs. In particular, both levels of representation share a notion of predicate-argument-adjunct structure. We have adapted the GPA to output DSyntS. We will illustrate its functioning using a simple example sentence, *Loyalty of local civilian officials is questionable.*

The output of Bilder is shown in Figure 4.3, with the projection from lexical items to their maximal projections indicated in boldface. The transformation of lexical items to nodes in the DSyntS is illustrated in Figure 4.4; the lexical items are morphologically analyzed and the information expressed morphologically is instead expressed by features. Further features may be added when the projection is followed from the lexical item to the maximal projection, as illustrated in Figure 4.5; here, the absence of a determiner leads to the addition of the

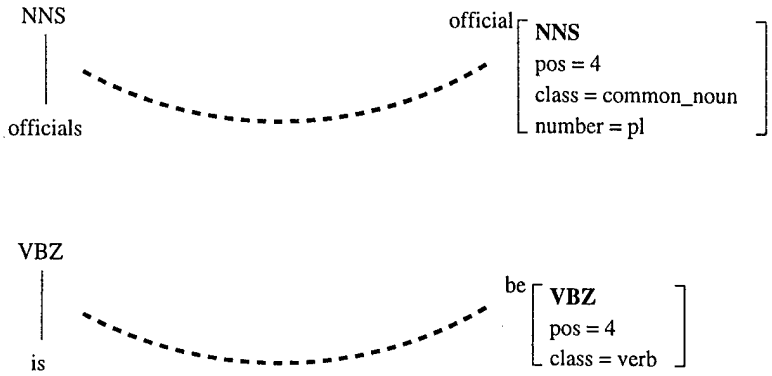


Figure 4.4: Local morphological analysis of lexical items, factoring inflectional markings into features

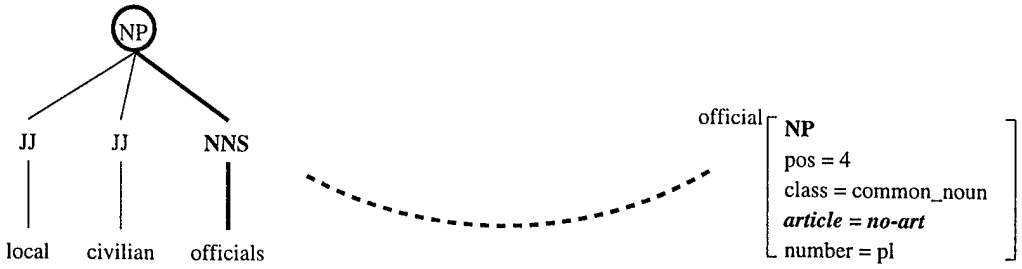


Figure 4.5: Additional feature marking on lexical head during propagation phase

feature `article:no-art`. Finally, argument and adjunct nodes are attached (Figure 4.6). Figure 4.7 shows an example, *response has been sent*, in which function words (in this case, the auxiliaries *have* and *been*) are expressed as features on the main lexical head.

4.2.3 Training on Corpora and Evaluation

We have trained Bilder on the Battlefield Message Corpus. The results are shown in Table 4.1.

We have also evaluated the combination of Bilder and the GPA against a "Gold Standard" annotated for deep-syntactic dependency relations (i.e., a DSyntS). We use a single score, **accuracy**, which corresponds to recall and is defined as the number of correct dependency arcs in the proposed parse divided by the number of dependency arcs in the Gold Standard parse.[2] On a test set of the Weather domain, Bilder (trained on the Penn Tree Bank) and the GPA attained an accuracy of 69%. This figure should not be compared directly to the corresponding figure for recall of constituents (which was also 69%), but the fact

[2]Because we require that the proposed parse be in fact a dependency tree, there is a maximum number of dependency arcs that the proposed parse can have. Therefore, there is no possible "trade-off" between recall and precision as in other areas, and precision is a relatively uninformative measure.

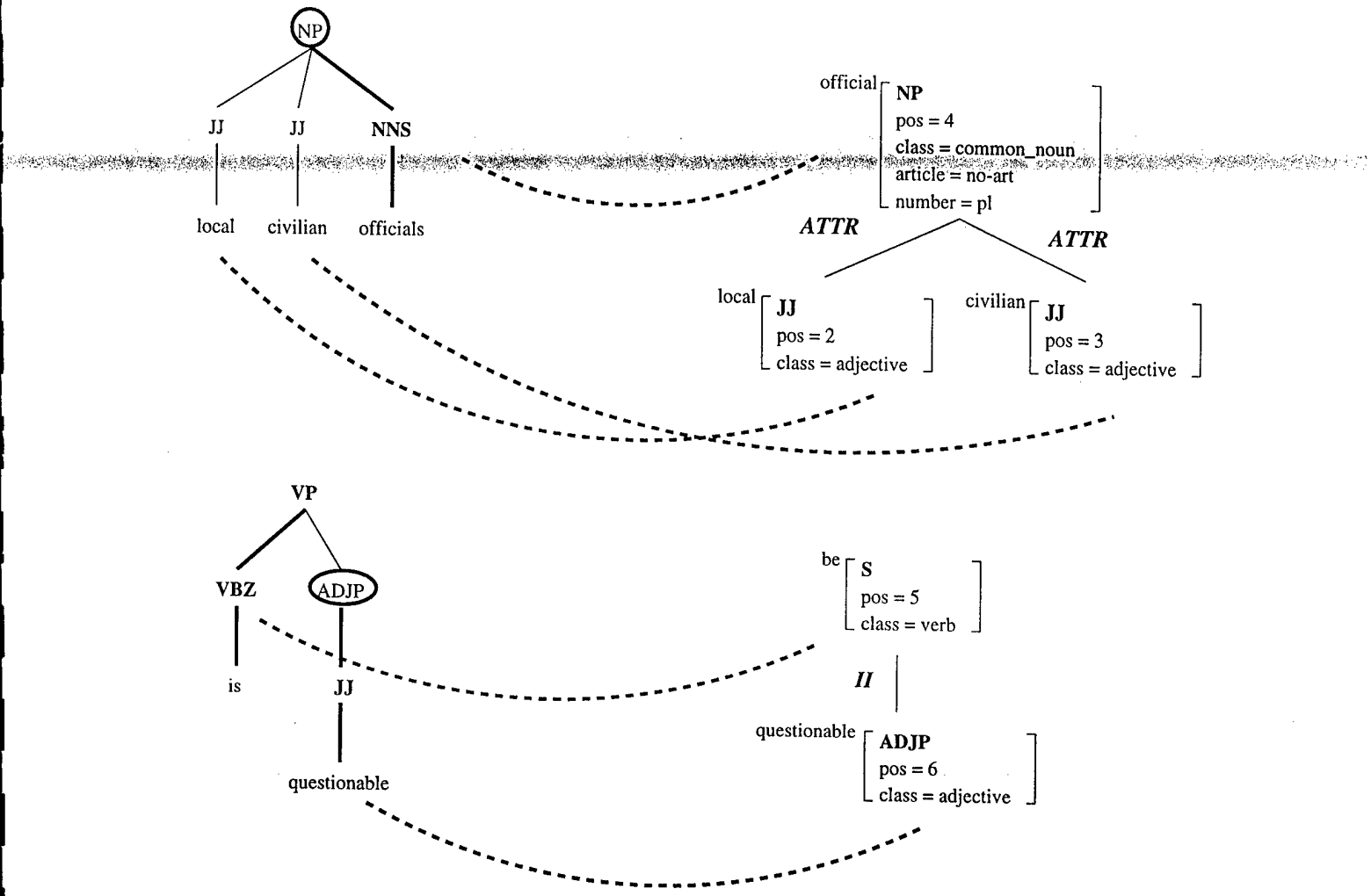


Figure 4.6: Attaching arguments and adjuncts to head

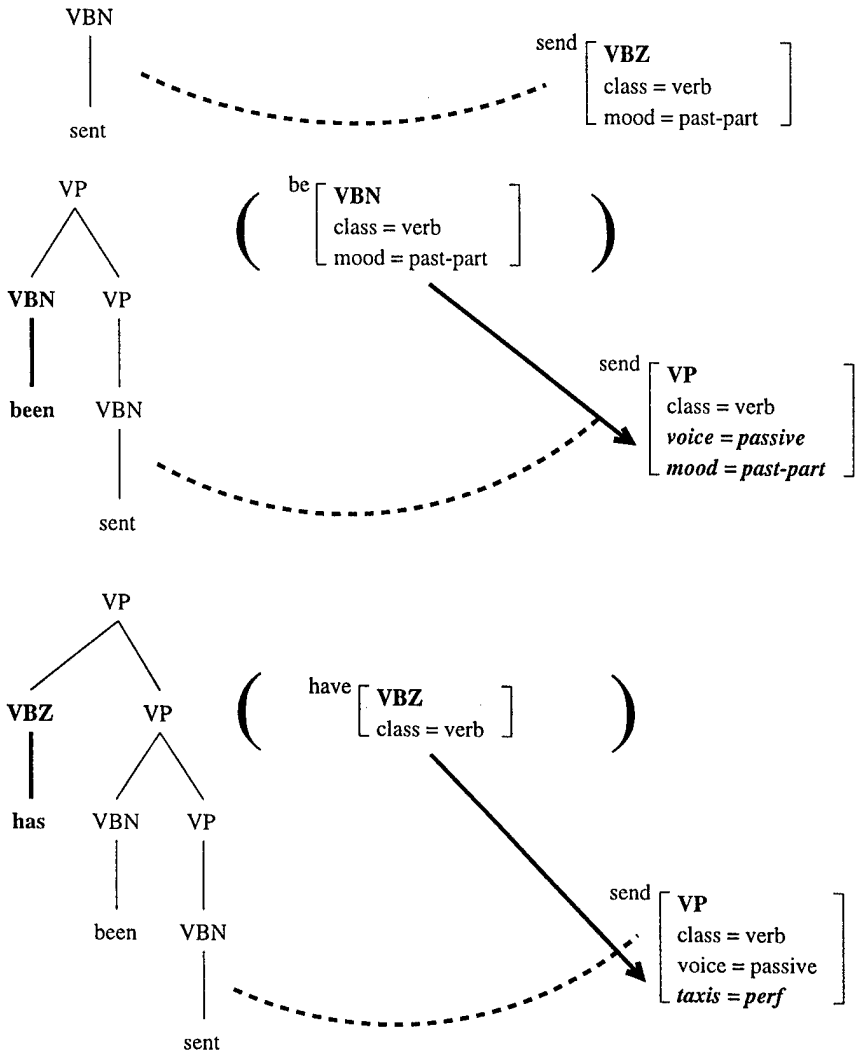


Figure 4.7: A more complicated example, showing absorption of functional lexical items (auxiliaries) as features for sentence *Response has been sent*

Model	LR	LP	CBs	0 CBs	≤ 2 CBs
Model 1	69.4	74.4	1.03	73.8	84.7
Model 2	72.4	77.1	0.93	76.7	85.7

Table 4.1: Model 1 was trained on WSJ treebank alone. Model 2 was trained on a mixture of WSJ treebank and domain-specific data. **LR/LP** = labeled recall/precision. **CBs** is the average number of crossing brackets per sentence. **0 CBs, ≤ 2 CBs** are the percentage of sentences with 0 or ≤ 2 crossing brackets respectively.

that there is no major decrease indicates that the GPA is doing a good job in assigning predicate-argument structures to the output of Bilder.

4.2.4 Discussion

The recall and precision achieved, even after training, on our corpus does not match the results obtained for the Wall Street Journal corpus. However, the training corpus we have used is tiny compared to the corpora usually used in statistical NLP: 415 sentences is a very small training set for what is a fairly broad domain, which is quite dissimilar from our existing treebank resources such as Wall Street Journal which contain tens of thousands of sentences. Given that its training was so restricted, Bilder performs surprisingly well. The most obvious way to improve Bilder's performance is therefore to annotate more training data. Furthermore, the new version of the parser (Collins, 1997) shows a 2.3% improvement over Bilder when tested on WSJ text — we hope that this improvement will carry across to the military messages. Closer inspection of the parser's output on the military messages domain is needed to identify other potential areas for improvement; the parser has been developed on Wall Street Journal text, and is naturally to some extent tuned to this corpus.

4.3 SuperTagging

SuperTagging is a technique developed at Penn to exploit the linguistically crafted grammar of English developed as part of the XTAG project (XTAG-Group, 1995). TAG, or Tree Adjoining Grammar, is a mathematical tree-rewriting formalism. The elementary structures are phrase-structure trees and they can be rewritten by two rewriting operations, adjunction and substitution. For more information, see (Joshi et al., 1991). XTAG is an implementation of this formalism in a graphical environment, which includes a CKY-style parser, and a broad-coverage grammar of English expressed in the TAG formalism.

4.3.1 TAG and Supertagging

The elementary trees of TAG localize dependencies, including long distance dependencies, by requiring that all and only the dependent elements be present within the same tree. As a result of this localization, a lexical item may be (and almost always is) associated with more than one elementary tree. We call these elementary trees *supertags*, since they contain more information (such as subcategorization and agreement information) than standard part-of-speech tags. Supertags for recursive and non-recursive constructs are labeled with βs and αs respectively.

The task of a lexicalized grammar parser can be viewed as a two step process. The first step is to select the appropriate supertags for each word of the input and the second step is to combine the selected supertags with substitution and adjunction operations. We call the first step *Supertagging*. Note that, as in standard part-of-speech disambiguation, supertagging could have been done by a parser. However, just as carrying out part-of-speech disambiguation prior to parsing makes the job of the parser much easier and therefore run faster, supertagging reduces the work of the parser even further.

More interesting is the fact that the result of supertagging is almost a parse in the sense that the parser need only link the individual structures to arrive at a complete parse. We present such a simple linking procedure (*Lightweight Dependency Analyzer*) in Section 4.3.4. This method can also be used to parse sentence fragments where it is not possible to combine the disambiguated supertag sequence into a single structure.

4.3.2 Trigram Model for Supertagging

The task of supertagging is similar to part-of-speech tagging in that, given a set of tags for each word, the objective is to assign the appropriate tag to each word based on the context of the sentence. Owing to this similarity of supertagging to part-of-speech tagging, we use a trigram model (Weischedel et al., 1993; Church, 1988) to disambiguate supertags. The objective in a trigram model is to assign the most probable supertag sequence for a sentence given the approximation that the supertag for the current word is only influenced by the lexical preference of the current word and the contextual preference based on the supertags of the preceding two words.

Although it is quite evident, owing to the rich information present in supertags, that the dependencies between supertags can easily span beyond the trigram context, one of the goals of this work is to explore the limits of the trigram tagging approach. It appears that a CKY style dynamic programming model that takes advantage of the dependency requirements of each supertag may perform better for supertag disambiguation. However, such an approach is too much like parsing and the objective here is to see how much disambiguation can be done without really parsing.

The lexical and contextual preferences are estimated from a corpus of sentences where the words are tagged with the correct supertag. The estimates for unseen events are arrived

at using a smoothing technique. We use Good-Turing discounting technique (Good, 1953) combined with Katz's back-off model for smoothing. We use word features similar to the ones used in (Weischedel et al., 1993), such as capitalization, hyphenation and endings of words, for estimating the unknown word probability. In conjunction with the word features, we exploit the organization of the supertags. The supertags are organized so that transformationally related supertags (indicative, passives, relative clauses, extraction supertags) are grouped into a single "family". Using this notion, if a word w_i in the training material appears with a supertag t_i which belongs to a tree family T, then w_i is associated with all the other members of the tree family T.

4.3.3 Experiments and Results

Table 4.2 shows the performance of the trigram model that was trained on two sets of Wall Street Journal data, 200K words[3] and 1000K words[4] and tested on 50K words[5]. The Treebank parses for the training and test sentences were converted into supertag representation using heuristics specified over parse tree contexts (parent, grandparent, children and sibling information). A total of 300 different supertags were used in these experiments. Supertag performance is measured as the percentage of words that are correctly supertagged by the model when compared against the supertags for the words in the test corpus.

Experiment 1: (Performance on the Wall Street Journal corpus) We used the two sets of data, from the XTAG parses and from the conversion of the Penn Treebank parses to evaluate the performance of the trigram model. Table 4.2 shows the performance on the two sets of data. The first data set, data collected from the XTAG parses, was split into 8,000 words of training and 3,000 words of test material. The data collected from converting the Penn Treebank was used in two experiments differing in the size of the training corpus; 200,000 words[6] and 1,000,000 words[7] and tested on 47,000 words[8]. A total of 300 different supertags were used in these experiments.

The dependency information encoded in the supertagger can be used in conjunction with a simple linking procedure as a robust, fast and efficient partial parser. Such an approach can also be used to parse sentence fragments where it is not possible to combine the disambiguated supertag sequence into a single structure.

[3]Sentences in wsj_15 through wsj_18 of Penn Treebank.
[4]Sentences in wsj_00 through wsj_24, except wsj_20 of Penn Treebank.
[5]Sentences in wsj_20 of Penn Treebank.
[6]Sentences in wsj_15 through wsj_18 of Penn Treebank.
[7]Sentences in wsj_00 through wsj_24, except wsj_20 of Penn Treebank.
[8]Sentences in wsj_20 of Penn Treebank.

Data Set	Size of training set (words)	Training	Size of test set (words)	% Correct
Converted Penn Treebank Parses	200,000	(Baseline)	47,000	75.3%
		Trigram	47,000	90.9%
	1,000,000	Unigram (Baseline)	47,000	77.2%
		Trigram	47,000	92.2%

Table 4.2: Performance of the supertagger on the WSJ corpus

4.3.4 The LDA

Supertagging associates each word with a unique supertag. To establish the dependency links among the words of the sentence, we exploit the dependency requirements encoded in the supertags. Substitution nodes and foot nodes in supertags serve as slots that must be filled by the arguments of the anchor of the supertag. A substitution slot of a supertag is filled by the complements of the anchor while the foot node of a supertag is filled by a word that is being modified by the supertag. These argument slots have a polarity value reflecting their orientation with respect to the anchor of the supertag. Also associated with a supertag is a list of internal nodes (including the root node) that appear within the supertag. Using the structural information coupled with the argument requirements of a supertag, a simple algorithm such as the one below provides a method for annotating the sentence with dependency links.

Step 1: `For each modifier supertag` s `in the sentence`
`Compute the dependencies for` s
`Mark the words serving as complements as unavailable for step 2.`

Step 2: `For the non-recursive supertags` s `in the sentence`
`Compute the dependencies for` s

Compute Dependencies for s_i of w_i:
`For each slot` d_{ij} `in` s_i `do`
`Connect word` w_i `to the nearest word` w_k `to the left or right of` w_i `depending on the direction of` d_{ij}`, skipping over marked supertags if any, such that` $d_{ij} \in$ `internal_nodes(`s_k`)`

An example illustrating the output from this algorithm is shown in Table 4.3. The first column lists the word positions in the input, the second column lists the words, the third lists the names of the supertags assigned to each word by a SuperTagger. The slot requirement of each supertag is shown in column four and the dependency links among the words, computed

by the above algorithm, is shown in the fifth column. The * and the . beside a number indicate the type of the dependency relation, * for modifier relation and . for complement relation.

Position	Word	Supertag	Slot req.	Dependency links
0	The	β_1	+NP*	2*
1	purchase	β_2	+N*	2*
2	price	α_2	–	
3	includes	α_{11}	–NP. +NP.	2. 6.
4	two	β_3	+NP*	6*
5	ancillary	β_4	+N*	6*
6	companies	α_{13}	–	

Table 4.3: An example sentence with the supertags assigned to each word and dependency links among words

4.3.5 Conversion to DSyntS

The output of the LDA, patterned after the derivation trees in TAG, differs from the DSyntS in several ways. CoGenTex has constructed a rule-based converter, which converts the LDA output to the DSyntS format.

Specifically, the following issues are handled:

- In predicative constructions involving a copula or similar verb (*Loyalty is questionable* or *Supplying fuel seems to be a problem*), the LDA will choose the predicate (*happy* or *problem*) as root of the tree, while the DSyntS requires the verb (*be* or *seem*) to be the root.
- In the DSyntS, but not in the LDA output, all function words (in particular determiners and auxiliaries) are represented not as separate nodes, but as combinations of features.
- In the LDA output, sentential arguments dominate the clause they are dependent on syntactically. The DSyntS reflects syntactic dependency directly.

A sample rule (for absorbing the perfective auxiliary *have* and representing it by the `taxis` feature on the main verb) is shown in Figure 4.8.

4.3.6 Training on Corpora and Evaluation

We have evaluated the SuperTagger that was trained on 200,000 word-supertag pairs of Wall Street Journal (WSJ) corpus on the 850 words of weather corpus and found that

```
CONVERSION-RULE:
[($V ATTR $AUX)]          |       [($V [mood:past-part])
                                   ($AUX [lexeme:have mood:?m])]
<-->
[$V]                      |       [($V [taxis:perf mood:?m])]
```

Figure 4.8: Sample conversionn rule to handle perfective auxiliaries in English

the SuperTagger performed at 87% accuracy, and on the 1,500 words of the battlefield message training corpus, where is performed at 86% accuracy.[9] We then retrained the SuperTagger on the 1500 word training corpus from the weather domain alone. On evaluating this SuperTagger on the 850 word test corpus, we found it to be only 78% correct. Finally, we retrained the SuperTagger on a combined corpus of 200,000 WSJ words and 5,000 battlefield message training words, and the accuracy improved to 89%.

We also evaluated the SuperTagger-LDA-converter combination against the test set of DSyntSs of the weather corpus, and obtained an accuracy of 65% (see Section 4.2.3, page 25 for an explanation of this score).

4.3.7 Discussion

As we have seen, the SuperTagger's performance on the weather test corpus decreased from 87% to 78% when trained on the weather training corpus. The decrease in performance is due to the small size of the training corpus in the weather domain. A better solution is to combine the domain corpus with the WSJ corpus so as to exploit the idiosyncrasies of the domain without compromising the high frequency information present in large corpora (which presumably are domain independent). The success of this approach is shown by our experiment using a combination of the battlefield corpus with the WSJ: the score increases from 86% to 89%. However, since the ratio of combining the domain specific knowledge to the WSJ knowledge is 1-1 this improvement is only 3% points. Other ways of combining the corpora (say, by adding the smaller corpus several times) may yield further improved results.

In addition, the LDA and the DSyntS-Converter currently have certain limitations. The LDA will occasionally propose analyses that are not *projective*, i.e., in which the diagram of the dependency analysis would have crossing arcs. While this occasionally occurs in natural language, the analyses are incorrect in the present cases, and the performance of the LDA would improve if it avoided non-projective analyses.

[9]The slight difference in numbers for the size of the training corpora given here is due to different pre-tokenization.

4.4 Limits of Stochastic Parsing

Given the small size of the training data, we are delighted with the performance of our parsers, and it is clear that our approach has been well validated. Further training and fine-tuning to the vocabulary will result in an even better performance. However, there are certain parsing difficulties that in a finite domain can be best overcome by access to particular types of well-understood semantic and pragmatic information (Palmer, 1990; Palmer et al., 1993). Simple subcategorization frame information with some selectional restrictions on verb arguments could help us avoid wrong parses. In addition, there are some particular pragmatic issues that have to be addressed to improve the quality of the translation.

One of the most obvious gaps in our current implementation is the ability to reference noun and verb class information. For example, Bilder misanalyzes *As the cloud thickens this evening showers will commence* by treating *thickens* as the main verb and *this evening showers will commence* as a sentential complement of *thickens*. Specific knowledge of which verbs can take sentential complements would preclude this, since *thicken* does not belong to that class. Additionally, ontological information about the noun phrase *this evening* could favor an analysis where it is attached as a modifier of the verb, rather than of the noun *showers*.

In the analysis of *The 175tr/9gtd is moving west on e4a48 Autobahn toward Alsfeld*, Bilder treats *west* as an argument of *move*. *Move* can be transitive, so the only way to rule this out is to recognize *west* as a directional phrase which is more properly an adjunct rather than a verb argument. In addition, we need to know that *move* is a *motion* verb with a directional phrase in order to generate the correct preposition for French, *vers* rather than *à*. (The corresponding sentence in the French corpus is *le 75 rc/9dcd se déplace vers l'ouest sur e4a48 autobahn vers alsfeld.*) The translation in Figure 2.1 was generated using a specialized entry in the transfer lexicon which treats *move west* as an idiom. While such entries are easy enough to add to the transfer lexicon, it is clear that a generalization is being missed.

Another weakness relates to adjunct attachment errors. For example, the sentence *Request you transmit via Maneuver Control System Fragmentary Order 5* is analyzed by Bilder with *transmit* as an intransitive verb, and *Maneuver Control System Fragmentary Order 5* as a complex noun phrase (complex noun phrases are very common in this domain). However, in this domain *transmit* is rarely used intransitively, the direct object of *transmit* should be an entity of type message such as *Fragmentary Order 5*, the *via* prepositional phrase should be a conduit of transmission such as *Maneuver Control System*, and a conduit of transmission cannot modify an entity of type message. Using this domain-specific knowledge about language use, the parser could easily identify the main verb *transmit* as having a *via* PP and a postposed direct object, *Fragmentary Order 5*.

Additional problems arise from the need to further interpret the syntactic analysis, in particular for dismabiguating pronouns (empty and overt). These problems also can profit from verb class information and from domain modeling. We return to this issue in Section 5.4.2.2.

Discourse and domain models will also be needed, as dicussed in more detail below, for the

appropriate translation of English pronouns into French, which often depends on whether the referent is masculine or feminine. In addition, the simple past tense in English has two different translations in French, depending on whether the situation being described is an event or a state, which will require an analysis of tense and aspect.

Chapter 5

The Core Transfer Component

In this chapter, we discuss the core of the translation system, the transfer component. All work presented in this section (with the exception of SABLE discussed in Section 5.3) was performed entirely during this project.

We start out by presenting a workbench for creating transfer lexicons (Section 5.1). We then present the transfer lexicon formalism in Section 5.2 and show how it can handle a broad range of cases. We discuss the techniques we use for automatically extracting translation lexicons from bilingual corpora (Section 5.3). Finally, we evaluate the transfer component (Section 5.4).

5.1 A Workbench for Transfer Lexicon

TransLex can draw on several separate transfer lexicons contained in separate files. These transfer lexicons are represented in an easily readable format, the Multi Lexical Base format (MLB in the figure). This format is used in several ways. First, the output of the automatic bilingual lexicon extractor (SABLE) is converted into MLB format. The resulting file can be hand-edited by a linguist or domain specialist. Second, additional MLB files containing translation lexicons can be hand-crafted, or re-used from other related or even unrelated domains. The MLB files are ordered so that in case of multiple occurrence of a key, the different entries for that key are ranked. Finally, the MLB files are automatically processed by the module *lextractor* to generate a fast loadable version of the transfer rules, sorted into a transfer lexicon (rules with lexical items) and a transfer grammar (rules without lexical items). Both of these are used by the actual transfer engine. Figure 5.1 depicts the entire process.

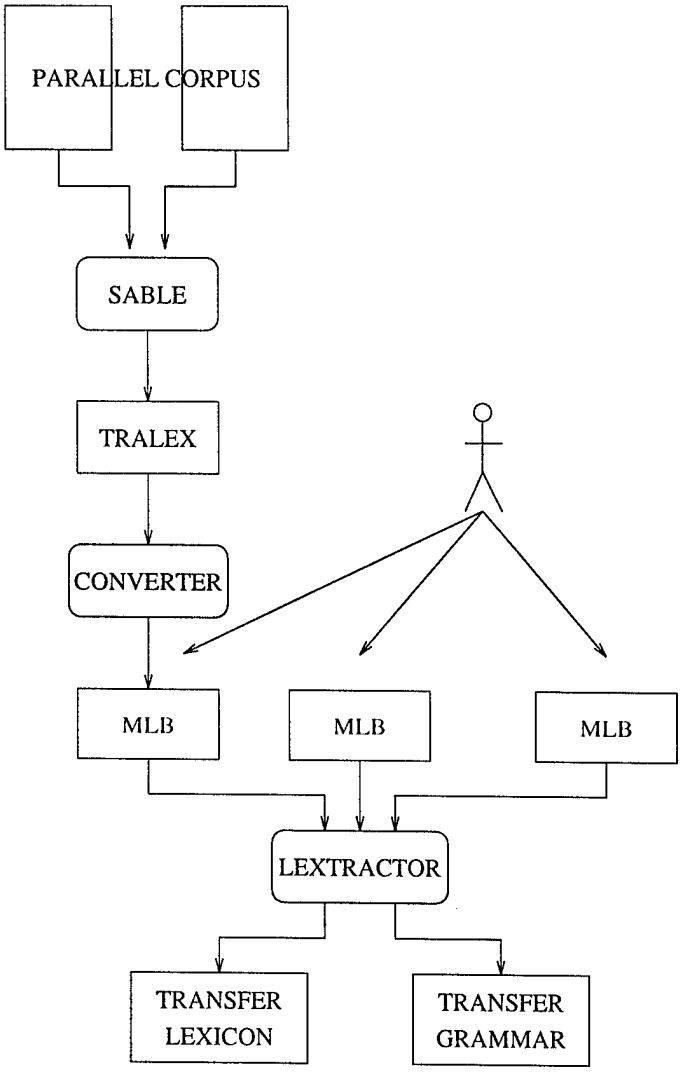


Figure 5.1: The transfer lexicon workbench

5.2 A Formalism for Transfer

In this section, we present the format we use for expressing transfer rules. We will first, in Section 5.2.1 propose a simple rule format, mapping a single lexical entry into another. Such rules cannot account for some phenomena described as *divergences* in (Dorr, 1994). These divergences will be presented in Section 5.2.2 and an extension of transfer rules format introduced, leading to complex rules. A quick discussion of the complex rules will conclude the section.

The rule format proposed here can be compared with Synchronous Tree Adjoining Grammars since it defines relations between two elementary trees by means of links between the nodes of the trees. The difference being that in our case the linked trees are dependency trees and not phrase structure trees.

The rule format proposed here is exactly the same as that used for transfer form DSyntS to SSyntS (surface syntactic structure) in RealPro, the generator. Therefore, the mechanism for interpreting rules such as the following could be reused.

5.2.1 Simple Rules

A simple MLB rule defines a mapping between two simple lexical items. The following rule relates the English lexeme *receive* to the French lexeme *recevoir* and to the German lexeme *empfangen*.

```
@LEX-ENTRY @EN receive
           @FR recevoir
           @GE empfangen
```

As we can see, an MLB is not specific to a single language pair, and is also non-directional. The Lextractor will automatically extract a language-pair and direction-specific translation lexicon and grammar for processing. In the following, we will sometimes represent such rules using a double arrow:

```
RECEIVE
<-->
REÇEVOIR
```

When applying such a rule on a DSyntS, the nodes that are not represented in the rule will remain unchanged after application of the rule. This is the reason why the arguments of the verbs have not been represented in the rule.

The rule presented above can account for the translation of the verb *receive* in the following example.

We have **received** your report $\longrightarrow$ Nous avons **reçu** votre rapport

It can be noted that, thanks to the lack of representation of governed preposition at the Deep Syntactic level, divergences concerning such prepositions ("structural divergences" in (Dorr, 1994)) do not require the introduction of special mechanisms. In the following example, the translation of the preposition *to*, into the French preposition *de* 'of' instead of *à* 'to' is realized implicitly at a monolingual level, when a DSyntS is transformed into a Surface Syntactic Structure.

The enemy **attempts to** flee $\longrightarrow$ L'ennemi **tente de** fuir ('the enemy tries of flee')

The prepositions introducing the actants of the verb *to attempt* and *tente* are represented in their respective monolingual lexical entries:

```
LEXEME:      ATTEMPT
CATEGORY:    verb
FEATURES:
GOV-PATTERN:[
DSYNT-RULE:
        [ ( ATTEMPT  II  X2 ) ]          | [ ]
        <-->
        [ ( ATTEMPT completive1 TO )
          ( OF prepositional X2 ) ]  | [ ]
]
MORPHOLOGY:  [
      ( []  attempt  [ reg ] )
     ]

LEXEME:      TENTER
CATEGORY:    verb
FEATURES:    [ aux:AVOIR ]
GOV-PATTERN:[
DSYNT-RULE:
        [ ( TENTER  II  X2 ) ]          | [ ]
        <-->
        [ ( TENTER completive1 DE )
          ( DE prepositional X2 ) ]  | [ ]
]
MORPHOLOGY:  [
      ( []  tenter  [ reg ] )
     ]
```

5.2.2 Complex Rules

There are many cases in which the translation of a sentence implies some changes in the structure of the sentence. The simple rules defined above cannot perform these structural modifications. (Dorr, 1994) has proposed a classification of those cases which are looked at as *divergences*. We borrow the classification proposed by her, but we introduce some modifications, and we (crucially) introduce some new types of divergences, namely "lexical divergence" and "lexical-collocational divergence". We also propose some extensions of the rule format presented above in order to take these divergences into account.

5.2.2.1 Thematic divergence

In a thematic divergence, a semantic actant of a verb is syntactically realized differently in the two languages. In the following example, the theme of the sentence is realized as the verbal object in English (*this place*) but as the subject of the verb in French (*cet endroit*).

The brigade **likes** this place $\longrightarrow$ Cet endroit **plaît à** la brigade

Such a divergence is handled by the following transfer rule, in which the Ist and IInd actants (subject and object, respectively) of the left hand part are permuted in the right hand part. (Recall that I designates the (syntactic) subject, II the object, and III the indirect object of a verb.)

```
LIKE (I  X
      II Y)
<-->
PLAIRE (I   Y
        III X)
```

This rule is in fact equivalent to the following tabular representation:

	Subject	Object
like	1	2
plaire_à	2	1

5.2.2.2 Promotional / Demotional divergence

Promotional and demotional divergences are two types of head switching divergence, where a head and one of its dependents are permuted. The following example exhibits a promotional divergence: the modifier (*almost*) is realized as an adverbial in English but as the main verb in French (*faillir*). We will say the adverbial has been *promoted* to become the main verb.

It **almost** rained. $\longrightarrow$ Il **a failli** pleuvoir.

Such an example can be handled by the following rule which promotes the adverb *almost* (Y) to become the French verb *faillir* and demotes the verb (X) which becomes the IInd actant of the new verb.

```
X [class:verb] (ATTR ALMOST)
<-->
FAILLIR (II X [mood:inf])
```

5.2.2.3 Conflational Divergence

A conflation occurs when a simple predicate is translated into a structure composed of a predicate and its necessary participants (or arguments). In the following example, the English verb *break-into* is translated by the verb-object structure *forcer l'entrée* in French. The semantic load of the English verb is split onto the verb and the direct object in French.

John **broke into** *Mary's room*
Jean **a forcé l'entrée** *de la chambre de Marie*
John has forced the entry to Mary's room

The incorporation or removal of a new participant is performed by a transfer rule which exhibits in one part a two node structure but a single node structure in the other.

```
TRANSFER-RULE:

BREAK-INTO (II  Y)

<-->

FORCER (II ENTRÉE [det:def]
        (II Y))
```

In some conflational cases, the semantic load is mainly carried by the direct object, leading to a light verb structure. Such a structure appears in the translation of the French verb *se suicider*, translated in English by the light verb structure *commit suicide*:

(1) a. *John* **commited suicide**

b. *Jean* **s'est suicidé**
John has suicided himself

Such a case will be taken into account by the following rule:

```
TRANSFER-RULE:

COMMIT (II SUICIDE)

<-->

SE-SUICIDER
```

5.2.2.4 Categorial Divergence

A categorial divergence appears when a predicate is realized by words of different category in French and English. In the following example, the English adjective *hungry* is translated the noun *faim* in French:

(2) a. *John* **is hungry**

b. *Jean* **a faim**
John has hunger

This divergence can be implemented by the following transfer rule which replaces the English adjective with the French noun and replaces the English copula by the verb *avoir* (*have*).

```
TRANSFER-RULE:

BE (II X [lexeme:HUNGRY])

<-->

AVOIR (II X [lexeme:FAIM])
```

5.2.2.5 Idioms

Idioms are often not translated literally between French and English. An idiom in the first language can be translated by another idiom or a simple expression in the second language. In the following example, the English idiom *kick the bucket* is translated in French by the idiom *se casser la pipe* (literally *break one's pipe*).

(3) a. *John* **kicked the bucket**

b. *John* **s'est cassé la pipe**
John broke his pipe

Translation of idioms are realized by multilexemic rules which map a multilexemic node into another.

```
TRANSFER-RULE:

KICK (II BUCKET [det:def])

<-->

SE-CASSER (PIPE [det:def])
```

It can be noted that such rules are actually simple rules featuring multilexemic nodes. It is an open question whether idiomatic expressions should be recognized before, during or after parsing, or during transfer.

5.2.2.6 Lexical Collocation Divergence

A collocation between two words X and Y in a language does not necessarily exist between the literal translations of X and Y in another language. In the following example, the English adjective *heavy* is translated into the adjective *fort* (literally *strong*) in French, in the collocative context *heavy rain.*

(4) a. *A* **heavy rain**

b. *Une* **pluie forte**
A strong rain

Such an example can be taken into account by the following rule which maps the two lexeme English structure *heavy rain* into the French structure *pluie forte.*

```
TRANSFER-RULE:

RAIN (ATTR HEAVY)

<-->

PLUIE (ATTR FORT)
```

5.2.2.7 Lexical Divergence

A lexical divergence appears when one word can be translated into two others depending on the context. The translation of the verb *lose* into Korean features such a divergence. When the direct objet of *lose* is a physical object, it is translated as *pwunsilhayssta*:

(5) a. *We* **lost** *the report.*

b. *Ku pokoselul* **pwunsilhayssta.**
'The report lost.'

But when the object is an event, *lose* will be translated as the verb *cyessta*:

(6) a. *We* **lost** *the battle.*

b. *ku centhwueyse* **cyessta.**
'The battle lost.'

This divergence can be handled by associating two rules to the English verb *lose*, each rule imposing a constraint on the semantic nature of the direct object:

1.
```
TRANSFER-RULE:

LOSE (II Y [type:physical_object])

<-->

PWUNSILHAYASSTA (II Y)
```

2.
```
TRANSFER-RULE:

LOSE (II Y [type:event])

<-->

CYESSTA (II Y)
```

A different type of lexical divergence involves not the semantic type of the argument of a verb, but its syntactic realization. Consider the following translations of the English verb *move*:

- Cloud will move into the western regions
  $\longrightarrow$ Des nuages **envahiront** les régions ouest
- A disturbance will move north of Lake Superior
  $\longrightarrow$ Une perturbation **se déplacera** au nord du lac supérieur
- The 79 dcg moves forward
  $\longrightarrow$ La 79 dcg **avance** vers l'avant
- They moved the assets forward
  $\longrightarrow$ Ils ont **amené** les ressources vers l'avant

These can be handled by having four entries in the MLB for *move*, differentiated by the argument structure:

```
@EN: move [class:verb]
       (ATTR into [class:preposition]
               (II $A))
@FR: envahir [class:verb]
       (II $A)

@EN: move [class:verb]
       (ATTR forward [class:adverb] )
@FR: avancer [class:verb]
       (ATTR vers [class:preposition
               (avant [class:noun]))

@EN: move [class:verb]
       (I $A
        II $B)
@FR: amener [class:verb]
       (I $A
        II $B)

@EN: move [class:verb]
@FR: déplacer [class:verb refl:+]
```

5.3 Automatically Extracting Transfer Lexicons

5.3.1 SABLE

SABLE, a new system for analyzing bilingual corpora (or "bitexts"), has recently been developed at Penn (Melamed, To appear). The greedy nature of the underlying algorithm,

the Smooth Injective Map Recognizer (SIMR), makes it independent of memory resources, and it is also able to allow crossing correspondences to account for word order differences. It does not require the two languages to use the same alphabet.

A critical application of SABLE for machine translation is the induction of domain-specific bilingual transfer lexicons (Resnik and Melamed, 1997). We have designed a fast algorithm for estimating a partial translation model, which accounts for translational equivalence only at the word level. A translation model is a set of transfer pairs, consisting of one word from each language which are (in some context in the bitext) a translation of one another. The model's accuracy/coverage trade-off can be directly controlled via a threshold parameter. (By setting the threshold lower, more transfer pairs are proposed, but fewer of these are likely to be correct.) This feature makes the model more suitable for applications that are not fully statistical. The model's hidden parameters can be easily conditioned on information extrinsic to the model, providing an easy way to integrate pre-existing knowledge such as part-of-speech, dictionaries, word order, etc.

The availability of such automatic tools greatly enhances our ability to quickly build transfer lexicons for special purpose domains. At this stage an automatically induced bilingual lexicon will not contain the detailed structural correspondences necessary for natural language generation in the target language, but it can certainly relieve us from the tedium of finding literal correspondences by hand. It can also provide a much needed bootstrapping level of mapping, which we can use to quickly pinpoint the phrases and constructions that will require more analysis.

In the presentation of the results, we will be using the following terminology. The **accuracy** is computed by dividing the number of correct transfers (determined manually) by the number of pairs in the transfer lexicon, while the **coverage** is computed by dividing the number of correct transfer pairs by the number of correct pairs needed to translate the text. In the following, **Backup** refers to a technique in which the results from the Hansards (Canadian bilingual record of parliamentary debates) are used in cases in which there is not enough data in the sublanguage corpus. The technique allows us to use data from the sublanguage corpus if it is available, and to use data from the more general Hansards when no specific translation can be derived from the sublanguage corpus. **Cutoff** refers to the threshold parameter mentioned above. As can be seen from the results, it is possible to "trade off" coverage against accuracy by adjusting this parameter.

Cutoff	Backup	Coverage	Accuracy
1	No	67.2 %	65 %
1	Yes	73 %	83 %
2	No	29.6 %	86 %
2	Yes	32.3 %	91 %

5.3.2 Discussion

The most useful results appear to be those obtained with a cutoff of 1 and backup. Some additional work needs to be done by hand to complete the translation lexicon, but this is facilitated by the workbench for transfer rules (see Section 5.1), and by the fact that the results reported above are quite good, given the small size of the corpus.

5.4 Evaluation of the Transfer Component

Contrary to the word-to-word automatically extracted transfer lexicon (*simple transfer lexicon* hereafter) and the parser, the linguistic ressources used by the transfer component are partially hand crafted through the enrichment of the simple transfer lexicon by complex rules. Quantitative evaluation is therefore difficult.

The evaluation of the transfer component has been performed in the following way: twenty five consecutive sentences have been chosen in the weather report corpora, they were parsed and the output DSyntS hand-corrected. These structures were first transfered to French DSyntS using the simple transfer lexicon (largely generated by SABLE) and realized as French sentences. As foreseeable, the result obtained using a word to word lexicon were poor and 41 complex rules were added to deal with certain classes of mistranslations.

We will first describe the types of complex rules which have been added to the transfer lexicon, then describe the phenomena that could not be adequately accounted for with the current complex rules.

5.4.1 Complex Rules Added by Hand

Forty-one complex rules have been added for the translation of the twenty five English sentences. They can be classified in the following categories:

- Compound nouns

  English compound nouns are usually translated by syntactically different noun phrases in French. The most common case is the transformation of a noun-noun structure into a noun-preposition-noun structure, as shown in the following two examples:

  Ex:
  pressure area $\rightarrow$ *zone de pression*

```
@LEX-MAP @EN: area [class:common_noun]
              (ATTR $A [lexeme:pressure class:common_noun])
         @FR: zone [class:common_noun]
              (ATTR de
                (II $A [lexeme:pression det:zero class:common_noun]))
```

Quebec City → *Ville de Québec*

```
@LEX-MAP @EN: city [class:common_noun]
               (ATTR Quebec [class:proper_noun])
         @FR: ville [class:common_noun det:def]
               (ATTR de
                 (II Québec [class:proper_noun det:zero]))
```

- Prepositions

  Prepositions tend to have different translations in French depending on their context. The common translation of the English preposition *in* is the French preposition *dans*, as in the following example:

  clouds will move into the western regions **in** *the wake of the high pressure area*
  →*les nuages envahiront les régions de l'ouest samedi après-midi* **dans** *le sillage de la zone de haute pression*

  This translation is performed by the following simple rule:

```
@LEX-MAP @EN: in [class:preposition]
         @FR: dans [class:preposition]
```

  But in the following example, the same English preposition is translated by the French preposition *de*:

  An increase **in** *cloudiness*
  Un accroissement **de** *l'état nuageux*

  To deal with such translation divergences, a complex rule translates *in* by *de* when the preposition introduces an argument of the noun *increase*:

```
@LEX-MAP @EN: increase [class:common_noun]
                (II $1 [lexeme:in class:preposition])
         @FR: accroissement [class:common_noun]
                (II $1 [lexeme:de class:preposition])
```

  Note that this case could also be handled if we consider *in* and *de* prepositions strongly governed by the respective nouns, and therefore included in the entries for these nouns in their respective monolingual dictionaries. In that case, the English parser would identify the preposition *in* as strongly governed and would delete it from the DSyntS. The transfer would simply be from *increase* to *accroissement* (which both have a direct object labeled II), and the French generator would add the *de*. While this approach

makes the transfer easier, it does require coordination between the source and target grammars (both need to recognize roughly the same type of strongly governed prepositions).

A third case of translation of the English preposition *in*, this time by the French preposition *à* appears in the following example:

The weather **in** *Quebec*
→*Le temps* **à** *(au) Québec*

Such a case can be readily taken into account by the following rule:

```
@LEX-MAP @EN: weather [class:common_noun]
                (ATTR $1 [lexeme:in class:preposition])
         @FR: temps [class:common_noun]
                (ATTR $1 [lexeme:à class:preposition])
```

But the latter rule can lead to mistranslations, as in the following example where the English preposition *in*, although introducing a circumstantial complement of the noun *weather*, like in the preceding example, is translated by the preposition *dans*:

The weather **in** *the maritime provinces*
→*Le temps* **dans** *les provinces maritimes*

This example shows that the choice of the right preposition may depend on the governor of the preposition and/or on its dependent. A solution could be to introduce complex rules for every lexico-syntactic structure of the form *N in N*, *X in N*, or *N in X*.

- Phrasal verbs

  Phrasal verbs are usually translated by simple verbs is French. Such translations are implemented as complex rules transforming a two word English structure (a verb and a particle) into a simple French verb.

  Ex:

  push across → *envahir*

```
@LEX-MAP @EN: push [class:verb]
                (ATTR across [class:preposition]
                   (II $1))
         @FR: envahir [class:verb]
                (II $1)
```

  Note that this example is rather sublanguage-specific.

- Comparative adjectives

  All comparative adjectives in English, including those with morphological comparative forms, are translated in French by an adjective modified by the relative adverb *plus*. These translations are performed by lexical expansion rules:

  Ex:
  cooler → *plus frais*

```
@LEX-MAP @EN: cooler [class:adjective]
         @FR: frais [class:adjective]
                (ATTR plus [class:adverb])
```

  The preceding rule deals only with the translation of the comparative adjective *cooler* and there will be as many translation rules as comparative adjectives to translate. This situation is not satisfactory: the regularity of the comparative adjective translation should be captured by a more abstract rule, accounting for the translation of all the comparative adjectives. Such a rule requires the representation, in the source and target language lexicons, of the relation existing between a comparative adjective and the base form of this adjective.

- Other Cases of lexical expansion/contraction

  Other cases of lexical expansion/contraction can be found in:

  - adverbs:
    southeastward → *vers le sud-ouest*

```
@LEX-MAP @EN: southeastward [class:adverb]
         @FR: vers [class:preposition]
                (II sud-ouest [class:common_noun det:def])
```

  - complex prepositions:

    southwest of → *à le (au) sud-ouest de*

```
@LEX-MAP @EN: southwest [class:common_noun]
                (ATTR $1 [class:preposition lexeme:of])
         @FR: à [class:preposition]
                (II sud-ouest [class:common_noun det:def]
                   (ATTR $1 [class:preposition lexeme:de]))
```

  - determiners:

    all of → *tout*

The distribution of the complex rules added according to the classes defined above is represented in the following table:

Type of rule	Number	Percentage
Compound nouns	14	34
Prepositions	12	29
Phrasal verbs	3	7
Comparative adj.	3	7
Other exp/red	9	22
Total	41	100

5.4.2 Discussion

There are several areas that can be improved in the Core Transfer Component.

5.4.2.1 Conditional Translations

Frequently, the choice of correct translation of a word depends on the lexical context in which that word appears:

heavy tank $\longrightarrow$ char lourd 'heavy tank'
heavy rain $\longrightarrow$ pluie forte 'strong rain'

The choice of the French translation *forte* in the second case depends on the main noun (and the same word is used for related meteorological phenomena). In this sense, the translation of *heavy* is conditional on the noun it is modifying. We do not currently account for this kind of phenomenon in our Phase I system.

5.4.2.2 Communicative Structure and Anaphors

In the Phase I implementation, there is no facility for interpreting anaphors (pronouns and definite noun phrases). Anaphor resolution is a major problem in MT (Raskin, 1987), in particular when the source and target language do not have identical anaphoric systems. Consider, for example, the following discourse from Japanese (Walker et al., 1994, Example 1):

1. Taroo ga kooen o sanpositeimasita
   $\text{Taroo}_{\text{SUBJ}}$ park in walking-was
   Taroo was taking a walk in the park

2. Ziroo ga hunsui no mae de mitukemasita
   $\text{Ziroo}_{\text{SUBJ}}$ fountain's front in found
   Ziroo found *Taroo* in front of the fountain
3. kinoo no siai no kekka o kikimasita
   yesterday's game's $\text{score}_{\text{OBJ}}$ asked
   Taroo asked *Ziroo* the score of yesterday's game

What is striking is that in sentence (2) above, an MT system cannot simply map the Japanese zero pronouns to English zero pronouns, since the resulting sentence, **Ziroo found in front of the fountain* is ungrammatical, while insertion of dummy arguments is not satisfactory: *Ziroo found something or someone in front of the fountain.* There is a similar problem in translating into English from Korean, since Korean also has zero pronouns. In addition, in the Battlefield Message domain, dropped arguments also occur in English. For example, in *Please pass to 149 Brigade.* the direct object is missing. However, the missing argument (the message is what is being passed) can be easily inferred from a domain model or discourse model, or both.

Even if no empty pronouns are involved, a difference in the gender system between two languages can lead to wrong results from a too simple translation of pronominal forms:

1. I saw a cat, **it** was running. $\longrightarrow$ J'ai vu un chat, **il** était en train de courir.
2. I saw a mouse, **it** was running. $\longrightarrow$ J'ai vu une souris, **elle** était en train de courir.

It is not only in the parsing module that discourse effects are important: during generation discourse context also plays a role. First, there is the obvious problem of when to generate what sort of pronoun in the target language (see (Tutin and Kittredge, 1992) for an overview). If source and target language have different types of pronouns (for example, Korean has empty pronouns, while English only in very limited contexts), then it is not simply possible to transfer pronoun type. Second, there is a problem with syntactic "constructions" and word order. Consider the following chapter-initial example from German and its English translation.

1. Die Nachrichtenbrigabden (Chapter Title)
   the intelligence brigades
2. Wertvolle Hilfestellungen leisten die Nachrichtenbrigaden.
   $[\text{valuable assistance}]_{\text{ACC}}$ provide $[\text{the intelligence brigades}]_{\text{NOM}}$
   The intelligence brigades provide valuable assistance.
   # Valuable assistance, the intelligence brigades of commerce provide.

In German, the direct object is in sentence-initial position in an active voice sentence. In English, such a word order, while syntactically possible, would be extremely strange and would make it more difficult for the reader to understand the discourse. He or she would expect to find the construction in a different context (say, following a sentence "The supply brigades provide useless assistance"), and would no longer understand what the information is that the text is conveying. In fact, the best translation uses a different construction, namely the passive voice: *Valuable assistance is provided by the intelligence brigades.*

5.4.2.3 Tense and aspect

In Phase I, we did not have the resources to address the difficult problem of tense and aspect. We currently translate tense and aspect by means of a direct mapping of grammatical tense and aspect features between the source and the target languages. This solution is, however, unsatisfactory since a given form can be translated differently in different contexts. For example, the English simple past tense is sometimes translated using the French *passé composé* form and sometimes using the French *imparfait* form:

1. A ridge line orientated in a north south direction through Quebec city this morning resulted [**simple past**] in sunny conditions along the entire eastern half of the St-Lawrence Valley today. $\longrightarrow$ Une ligne de crête orientée ce matin du nord au sud au-dessus de Québec a apporté [**passé composé**] du soleil aujourd'hui tout le long de la moitié est de la vallée du St-Laurent.
2. Temperatures at 4 a.m. were [**simple past**] mostly in the twenties and low thirties. $\longrightarrow$ A quatre heures les températures étaient [**imparfait**] généralement de 20 à 34.

As is well known (Kamp and Reyle, 1993), one factor that correlates with the choice of *grammatical aspect* — in this case, the choice between the perfective *passé composé* form and the imperfective *imparfait* form — is the *aspectual type* (a.k.a. *Aktionsarten*) of the sentence in question. While conflicting theories of aspectual types abound, there is general agreement (at least informally) that sentences such as the two given above describe situations of different types, the first describing an event and the second a state; moreover, it is generally agreed that while the choice of grammatical aspect does reflect a choice of perspective, perfective forms are usually used for eventive sentences, and imperfective forms for stative sentences. Consequently, since the simple past in English is an unmarked form, the choice of the *passé composé* for the eventive sentence and the *imparfait* for the stative one is entirely expected here.

A similar (and equally common) problem exists in translating the French simple present form into English, where a choice must be made between the English simple present and the English progressive: in ordinary contexts, the choice is again determined by aspectual type, as stative French sentences in the simple present are usually translated using the English simple

present, while eventive sentences usually receive a translation using the English progressive. Finally, the choice of tense and aspect may interact with certain syntactic configurations. As an example, consider the use of the French future tense vs. the English present tense in the temporal subordinate clause:

> ... and as the cloud thickens [**present**] this evening showers will commence and continue in the western half of Quebec tomorrow.
>
> ... au fur et à mesure que le ciel s'assombrira [**future**] ce soir des averses commenceront à se produire et se poursuivront demain dans la moitié ouest du Québec.

Chapter 6

Generation

6.1 RealPro

For generation, we have used RealPro, CoGenTex's sentence realizer (Lavoie and Rambow, 1997). The input representation for RealPro is precisely the DSyntS formalism which we use for transfer. We have constructed a small French grammar during Phase I. We based this grammar on the English grammar, which we adapted through successive modifications. We list the main areas of syntax which required significant changes below in Section 6.2. The ease with which we were able to accomplish this port is a confirmation of the modular nature of grammars in RealPro, and also of the similarities between English and French. (Attempting generation in more diverse language such as Korean might require somewhat more effort, since fewer syntactic rules could be reused directly.) We used morphological components previously developed at the Université Paris 7, whose help in this matter we gratefully acknowledge.

6.2 French Generation Grammar

The new French grammar comprises rules for the following constructions and syntactic phenomena not found in English, or significantly different from their English counterparts.

- Agreement between adjectives and the nouns they are predicated of.
- Auxiliary insertion.
- Reflexive verbs (e.g., *se trouver* – 'to be located').
- Clitic pronouns.
- Determiner insertion.

As an example of changes needed for the grammar, we will show some rules to handle clitics. Clitics are pronominal forms which have markedly different syntactic behavior than the full NPs they replace in that they are closely bound to (usually) the finite verb of the sentence, and occur immediately before it. Therefore, the linear order of the main verb and the direct (or indirect) object depends on whether or not the direct (or indirect) object is pronominal or not. This rule is reflected in the linear order rule (i.e., surface-syntactic rule) shown in Figure 6.1. In that figure, the direct object is given the arc label COMPLETIVE1 and the indirect object is given the arc label COMPLETIVE2.

```
/------------------------ VERBS -------------------------
GLOBAL-RULE:

LEFT-DEPS:
(V      completive1     QUE)
(V      predicative     SUJ)
(V      restrictive-ne  NE)
(V      completive1     LE)
(V      completive2     LUI)

RIGHT-DEPS:
(V      restrictive     PAS)
(V      adverbial       ADV)
(V      completive1     OBJD)
(V      completive2     OBJI)

(V      completive3     CIRC)
(V      adverbial       PREP)
(V      adverbial       CONJS)
(V      adverbial       N)
(V      adverbial       V2)
(V      coordinative    CONJC)

CONDITIONS:
(V              [verb ~taxis:perf ~voice:pass])
(PREP           [preposition])
(LE             [clitique-objetd])
(LUI            [clitique-objeti])
(QUE            [pro-rel])
(ADV            [adverb])
(CONJS          [subordinative_conj])
(N              [noun])
(V2             [verb mood:pres-part ])
```

Figure 6.1: Linear order (i.e., surface-syntactic) rules for French, reflecting special role of clitics (pronominal forms of arguments and adjuncts)

Chapter 7

Porting TransLex

7.1 Porting to a New Domain

The system can be easily ported to new domains. The only task that must be performed is that the transfer lexicon may need to be augmented. This step may use Sable or some such automatic tool (using a bilingual corpus), but it will definitely require some human intervention as well.

In addition, the parser may be retrained on the domain to augment accuracy of parsing. This would require annotation of domain texts by linguistically-trained personnel.

7.2 Porting to a New Language Pair

During Phase I, we have ported our translation system to perform English-to-Arabic translations in the weather report domain in order to show that the framework easily allows for such porting. As data, we had an Arabic translation of the weather report corpus (translated by hand by a professional translator for this purpose). We performed the following tasks:

- We used Sable to extract a translation lexicon. This was done through an iterative approach: an initial translation lexicon, based only on the division of the two texts into corresponding paragraphs, yielded predictably bad results. However, a computational linguist (Arabic native speaker) could easily check this list and determine which proposed translations were correct. The reduced list, containing only correct translations, was subsequently used as a "seed" and Sable was invoked again. In addition, the linguist supplied a list of function words and their translation. These results were significantly better, and subsequently augmented by hand.
- We developed an Arabic grammar for RealPro. To do this, we started with the English grammar and incrementally modified it until all constructions present in the corpus

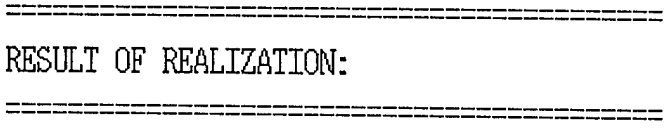


Figure 7.1: Arab translation generated from *Skies were clear across the three maritime provinces early this morning.*

were accounted for.

We did not implement a general processor for the complex Arab morphology, since off-the-shelf processors are available, for example from Xerox's InXight subsidiary. (And, of course, the issues which we did not address for French tranlsations, notably a discourse module for determiner generation and a tense and aspect module, also remain unimplemented for Arabic.) A sample output is shown in Figure 7.1.

We can generalize the porting process to the following procedure.

- If using the SuperTagger, create an XTAG grammar for the source language. If using Bilder, create a syntactic annotation scheme for corpora in the source language.
- Annotate monolingual corpus (with supertags if using the SuperTagger, with syntactic phrase-structure trees if using Bilder).
- Train parser (Bilder or SuperTagger).
- Devise a RealPro grammar for the target language.
- Run Sable or other bilingual translation lexicon extraction tools on bilingual parallel corpus.
- Using the workbench, augment the automatically generated translation lexicon by hand.

Chapter 8

Future Work

TransLex, the framework that we have developed during Phase I and which we have described in the previous chapters, has shown itself to be robust and adequate for the task. However, the system implemented in Phase I is only a demonstration prototype, and certain extensions to the framework must be made to turn it into a fully operational prototype: as we discussed throughout the preceding presentation of our Phase I system, certain phenomena are not handled presently, and furthermore, certain solutions are clumsy or complex. In this chapter, we discuss problems and sketch possible solutions to the problems and extensions to the system.

8.1 Possibilities for Improvement

We summarize here the main areas for improvement that we have identified in the preceding chapters.

- The parsers can be improved in certain ways discussed in Section 4.2.4, page 28, and in Section 4.3.7, page 33.
- The parsers currently lack certain types of knowledge that would enable them to make correct attachment decisions about arguments and adjuncts, as discussed in Section 4.4, page 34.
- Translations are often conditioned on lexical context in ways which are currently difficult to generalize. See Section 5.4.2.1, page 51.
- Pronouns (both empty and overt) cannot simply be translated literally as is done currently in TransLex; instead, their referent must be identified at least partially. See Section 5.4.2.2, page 51.

- Like pronouns, tense and aspect cannot be translated literally (as is done currently in TransLex), since different languages have different stocks of morphological forms and use them differently. See Section 5.4.2.3, page 53.

8.2 Sketch of a Proposed Extension to the Architecture

We propose to keep the framework developed in Phase I, but to extend it by adding some specific functionality not currently present and by improving currently available functionality. Specifically, the following functionality needs to be added:

- A module that can translate tense and aspect between languages.
- A discourse module that can translate anaphors and context-sensitive syntax between languages.
- A module that represents domain knowledge (both domain-specific lexical information and ontological information) for the battlefield message domain (and its subdomains) in a manner that it can be readily accessed by the parsers, the transfer component and the discourse module. This will aid in disambiguation tasks and further specification of underspecified representations.

Existing functionality needs to be improved in the following ways:

- The English parsers and the parsing frameworks need access to the domain knowledge.
- The notion of "lexical function" should be introduced into the transfer component, which will allow for a much more compact representation, and will allow the transfer component to access the domain knowledge.
- To aid in the sublanguage analysis, supertagging can be used to extend existing techniques for automatic extraction of domain terminology.
- We will study the practical aspect of using some limited information about sublanguages to sharpen the results obtained using the tools to process corpora which contain multiple topic areas.

Because of the modular "plug-and-play" architecture of our framework as developed in Phase I, each of the tasks can be worked on independently, and the results can be easily integrated into the framework.

The overall proposed revised architecture is shown in Figure 8.1. Components not yet designed or implemented, as well as corpora not yet annotated and knowledge bases not

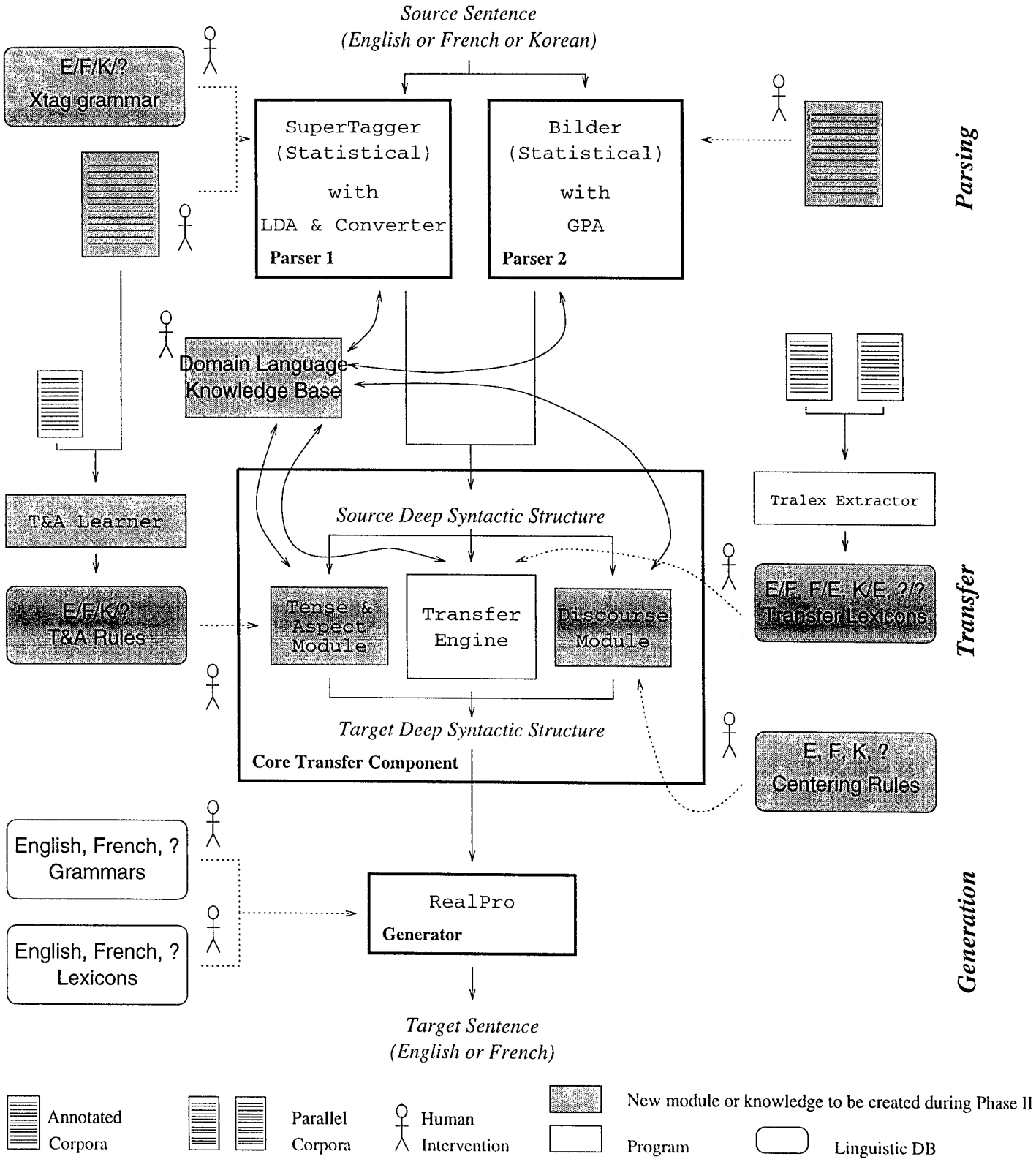


Figure 8.1: Proposed extension to architecture of MT system ("?" refers to new languages that can be added to the framework)

yet created, are shown in grey. The core transfer component and the other plug-and-play components (two parsers and the generator) are outlined by heavy black boxes. The needed knowledge bases are also shown, along with the tools for creating them (T&A Learner and Tralex Extractor). The stick figures show where human intervention is needed when the translation framework is to be ported to a new language and/or a new subdomain. (Of course, certain resources may be reusable.)

Bibliography

Bourbeau, L. (1991). Intelligent natural language interface — subtask 5: Sublanguage semantic pattern gathering. Technical report, Progiciels Bourbeau Pinard Inc. Final Report, US Army CECOM, Contract number DAAB07-89-D-A050.

Briscoe, T. and Carroll, J. (1993). Generalized lr parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19(1).

Church, K. W. (1988). A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *2nd Applied Natural Language Processing Conference*, Austin, Texas.

Collins, M. (1996). A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, CA.

Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain.

Collins, M. and Brooks, J. (1995). Prepositional phrase attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora.*

Dorr, B. J. (1994). Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–635.

E. Black et al. (1991). A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proceedings of the February 1991 DARPA Speech and Natural Language Workshop.* DARPA.

Good, I. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika 40 (3 and 4).*

Hindle, D. and Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1).

Hirschman, L., Palmer, M., Dowding, J., Dahl, D., Linebarger, M., Passonneau, R., Lang, F., Ball, C., and Weir, C. (1989). The pundit natural-language processing system. In *AI Systems in Government Conference.* Computer Society of the IEEE.

Jelinek, F., Lafferty, J., Magerman, D., Mercer, R., Ratnaparkhi, A., , and Roukos, S. (1994). Decision tree parsing using a hidden derivation model. In *Proceedings of the Human Technology Workshop*, pages 272–277.

Joshi, A. K. and Srinivas, B. (1994). Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing. In *Proceedings of the* 17^{th} *International Conference on Computational Linguistics (COLING '94)*, Kyoto, Japan.

Joshi, A. K., Vijay-Shanker, K., and Weir, D. (1991). The convergence of mildly context-sensitive grammatical formalisms. In Sells, P., Shieber, S., and Wasow, T., editors, *Foundational Issues in Natural Language Processing*, pages 31–81. MIT Press, Cambridge, Mass.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Kluwer Academic Publishers.

Lafferty, J., Sleator, D., and Temperley, D. (1992). Grammatical Trigrams: A Probabilistic Model of Link Grammar. Technical Report CMU-CS-92-181, School of Computer Science, Carnegie Mellon University.

Lavoie, B. and Rambow, O. (1997). RealPro – a fast, portable sentence realizer. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP'97)*, Washington, DC.

Magerman, D. (1995a). Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics.*

Magerman, D. and Marcus, M. (1991). Pearl: A probabilistic chart parser. In *Proceedings of the European Assoc. for Comp. Ling.*, Berlin.

Magerman, D. M. (1995b). Statistical Decision-Tree Models for Parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics.*

Marcus, M. M., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19.2:313–330.

Melamed, I. D. (To appear). Automatic Discovery of Non-Compositional Compounds in Parallel Data. In *Proceedings of the ACL-97*, Madrid, Spain.

Mel'čuk, I. A. (1988). *Dependency Syntax: Theory and Practice.* State University of New York Press, New York.

Palmer, M. (1990). *Semantic Processing for Finite Domains.* Cambridge University Press, Cambridge, England.

Palmer, M. (1996). Capturing semantics and pragmatics for translation. invited presentation, Interlingua Workshop at AMTA-96.

Palmer, M. and Rosenzweig, J. (1996). Capturing motion verb generalizations with synchronous tags. In *Proceedings of AMTA-96*, Montreal, Quebec.

Palmer, M., Weir, C., Passonneau, R., and Finin, T. (1993). The Kernel text understanding system. *Artificial Intelligence*, 63:17–68. Special Issue on Text Understanding.

Pereira, F. and Schabes (1992). Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics.*

Rambow, O. and Joshi, A. (1996). A formal look at dependency grammars and phrase-structure grammars, with special consideration of word-order phenomena. In Wanner, L., editor, *Current Issues in Meaning-Text Theory.* Pinter, London.

Raskin, V. (1987). Linguistics and natural language processing. In Nirenburg, S., editor, *Machine Translation: Theoretical and Methodological Issues.* Cambridge University Press.

Resnik, P. and Melamed, I. D. (1997). Semi-Automatic Acquisition of Domain-Specific Translation Lexicons. In *Proceedings of the ANLP-97*, Washington, D.C.

Shieber, S. and Schabes, Y. (1990). Synchronous tree adjoining grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki.

Srinivas, B. (1997). *Complexity of Lexical Descriptions and its Relevance to Partial Parsing.* PhD thesis, Computer Science Department, University of Pennsylvania.

Tutin, A. and Kittredge, R. (1992). Lexical choice in context: Generating procedural texts. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92).*

Walker, M. A., Iida, M., and Cote, S. (1994). Japanese discourse and the process of centering. *Computational Linguistics*, 20(2):193–233.

Weischedel, R., Schwartz, R., Palmucci, J., Meteer, M., and Ramshaw, L. (1993). Comping with ambiguity and unknown words through probabilistic models. *Computational Linguistics*, 19.2:359–382.

XTAG-Group (1995). A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS 95-03, University of Pennsylvania.